

Maarit Klami

IRC:n käyttö yritysten viestinnässä

Metropolia Ammattikorkeakoulu
Insinööri (AMK)
Tietotekniikan koulutusohjelma
Insinöörityö
3.5.2012

Tekijä Otsikko	Maarit Klami IRC:n käyttö yritysten viestinnässä
Sivumäärä Aika	34 sivua + 2 liitettä 3.5.2012
Tutkinto	insinööri (AMK)
Koulutusohjelma	tietotekniikka
Suuntautumisvaihtoehto	ohjelmistotekniikka
Ohjaaja	lehtori Peter Hjort
<p>Tässä työssä perehdyttiin IRC:n käyttömahdollisuuksiin yritysten sisäisessä ja ulkoisessa viestinnässä. Työssä verrattiin IRC:tä muihin pikaviestimiin sekä sosiaalisen median tarjoamiin viestintä- ja markkinointimahdollisuuksiin. Eräs tarkastelluista seikoista on yrityksen koon vaikutus siihen, kuinka mittava operaatio IRC:n käyttöönotto yrityksessä on. Työssä on esitetty myös mahdollisuuksia muiden pikaviestimien integroimiseksi IRC:hen. Työ on tehty mielenkiinnosta käyttää IRC:tä viestinnän apuna yrityksissä.</p> <p>Tietoturva on keskeisessä roolissa yritysten toiminnassa, josta johtuen asia on huomioitava myös viestintäjärjestelmien käyttöönotossa ja päivittäisessä käytössä. Työssä tietoturvaa on käsitelty sekä tekniseltä että inhimilliseltä kannalta.</p> <p>Työssä yhtenä keskeisenä osana on tutustuttu IRC-bottien hyödyntämiseen yritysten toiminnassa. Osana työtä tehtiin Eggdropilla IRC-botti, jolle perusominaisuuksien laajentamiseksi ohjelmoitiin TCL-ohjelmointikieltä käyttäen halutun toiminnallisuuden toteuttava skripti. Työssä esitellään botin käyttöönoton vaatimat toimenpiteet sekä skriptin toteuttaminen. Lisäksi työssä on esitelty erilaisia bottien toiminnallisuuksia ja niiden hyödyntämismahdollisuuksia yrityskäytössä. Botteja voidaan luoda ja muokata yritysten tarpeisiin sopiviksi hyvinkin yksityiskohtaisesti.</p> <p>Työssä todettiin IRC:n soveltuvan hyvin viestintään, joka suuntautuu yrityksestä ulospäin tai usealle työntekijälle yrityksen sisällä. Hankaluutana IRC:n käytössä pidettiin sen melko monimutkaista käyttöönottoa sekä käytön opettelua. Positiivisena asiana työssä pidettiin IRC:n tarjoamia laajoja mahdollisuuksia erilaisiin toimintoihin.</p>	
Avainsanat	IRC, botti, Eggdrop, pikaviestin, tietoturva

Author(s) Title	Maarit Klami Using IRC in business communication
Number of Pages Date	34 pages + 2 appendices 3 May 2012
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor(s)	Peter Hjort, Lecturer
<p>This thesis deals with the opportunities of IRC in communication both inside and outside business organizations. IRC is compared to both other instant messaging solutions and other modern communication and marketing opportunities provided by social media. One of the topics in question is the correlation between the size of a company and the effort required to introduce the IRC solutions. Also the integration of other instant messaging systems to IRC is covered.</p> <p>Information security is a key factor in the business world and, therefore, it must also be noted in the introduction and usage of messaging systems. This thesis approaches security from both a technical and a human point of view.</p> <p>One important part in the thesis is the usage of IRC bots. Part of this involves an Eggdrop IRC bot which is expanded by a customized script to implement wanted functionalities. The installation of the bot and the script is also dealt with. Furthermore, a variety of different script concepts with usage examples are presented. Bots can be thoroughly customized to suit the needs of an organization.</p> <p>In the thesis it is shown that IRC is well-suited for communication between great amounts of employers inside a company, or for communicating outside the organization. The hardest parts are seemingly the initialization and learning how to use IRC. Positive, however, is the variety of opportunities IRC offers.</p>	
Keywords	IRC, bot, Eggdrop, instant messaging, information security

Sisällys

1	Johdanto	1
2	IRC:n käyttö yritysten viestinnässä	1
2.1	IRC ja Viestintä	1
2.2	IRC	3
2.2.1	IRC:n historia	3
2.2.2	Asiakasohjelmat	4
2.2.3	IRC:n keskustelumuodot	6
2.2.4	IRC-verkon toiminta	6
2.2.5	Nimimerkkien käyttö	7
2.2.6	Käyttäjän tiedot	8
2.3	Hyödyt IRC:n käytössä yritysten viestinnässä	9
2.4	IRC:n heikkoudet muihin pikaviestimiin nähden	11
2.5	Erot pienissä ja suurissa yrityksissä	12
3	Tietoturva IRC:n käytössä yrityksen viestinnässä	13
3.1	Tietoturva	13
3.2	SSL-salausprotokolla	14
3.3	Sosiaaliset uhat	14
3.4	Viestintä yrityksen sisäällä ja yrityksen ulkopuolella	15
4	IRC-botti	16
4.1	IRC-bottien käyttö	16
4.2	Botin toteutus	17
4.3	Eggdrop	18
4.4	Eggdropin ominaisuudet	18
4.4.1	Perusominaisuudet	18
4.4.2	Partyline	19
4.4.3	Kanavienhallinta	20
4.4.4	Jononhallinta	21

4.4.5	Botnetit	22
4.5	TCL ja skriptit	22
5	Skriptien käyttö	23
5.1	Bottien hyödyntäminen	23
5.2	”Mitä tarvitsee tehdä”- skriptin esittely	24
5.3	Muita esimerkkejä botille tehtävistä skripteistä	27
5.3.1	Botin toiminta kanavan vartijana	27
5.3.2	Botti apuna ajanhallinnassa	28
5.3.3	Osoitekirja botille	28
5.3.4	Tiketöintijärjestelmän yhdistäminen bottiin	29
5.4	IRC:n laajentaminen muihin palveluihin	29
6	Yhteenveto	31
	Lähteet	33
	Liitteet	
	Liite 1. Eggdropin asetustiedosto	
	Liite 2. ”Mitä tarvitsee tehdä”- skripti	

Lyhenteet ja käsitteet

Asiakasohjelma

Asiakasohjelman kautta käyttäjät ottavat yhteyden IRC:iin esimerkiksi mIRC tai PuTTY.

Bitlbee Ohjelma, jonka avulla IRC:iin voi tuoda muiden pikaviestinten tai sosiaalisen median viestejä

Botti Ohjelma, joka toimii itsenäisesti sille määriteltyjen toimintaohjeiden mukaan. Botteja käytetään etenkin silloin kun tarvitaan suurta työmäärää, toistoa tai jatkuvaa päivystystä.

DCC *Direct client to client*, Tiedonsiirtomenetelmä

Eggdrop UNIX-järjestelmissä toimiva modulaarinen IRC-bottiohjelma.

Identti IRC-käyttäjän yksilöllinen tunniste.

IRC *Internet Relay Chat*, reaaliaikainen keskusteluohjelma.

IRC-verkko Useista IRC-palvelimista koostuva verkko, esimerkiksi IRCnet.

IRC-palvelin IRC-palvelimen avulla otetaan yhteys IRC-verkkoon.

Isäntänimi Käyttäjän tietokoneen nimi tai sen IP-osoite (engl. hostname).

Kanava Samanhenkisten käyttäjien keskustelualue.

Loki Tiedosto johon tallennetaan IRC-keskustelun viestit.

MediaWiki Avoimen lähdekoodin yhteisöllinen dokumentointialusta.

Nimimerkki Käyttäjän yksilöivä lempinimi IRC:ssä.

Operaattorin oikeudet

Antavat oikeuden valvoa kanavan järjestystä.

Partyline	Käyttäjän ja botin välinen ohjauskanava.
Pikaviestin	Ohjelma jolla voi keskustella internetissä reaaliajassa.
Puheoikeus	Antaa puheoikeuden kanavalle (engl. voice).
Python	Tulkattava ohjelmointikieli.
Rivitulva	Käyttäjä kirjoittaa lyhyessä ajassa useita rivejä tekstiä (engl. flood).
Shell-tunnus	Tunnus etäkäytettävään palvelimeen.
Skripti	Pieni ohjelma, joka on sisällytetty toisen ohjelman sisälle.
SSL	<i>Secure Sockets Layer</i> , IRC:ssä käytetty salausprotokolla.
TCL	Tulkattava ohjelmointikieli.
TCP/IP	<i>Transmission Control Protocol / Internet Protocol</i> , Internetissä pääsääntöisesti käytetty protokollayhdistelmä.
Tiketti	Työtehtävä
Tikettijärjestelmä	Työtehtävien ohjaus- ja hallintajärjestelmä.
UNIX	Single UNIX Specification-määritelmän mukainen käyttöjärjestelmä.
MS Windows	PC:lle tarkoitettujen graafisten käyttöliittymien ja -järjestelmien perhe.
Yksityisviesti	Käytetään silloin kun haluaa keskustella vain yhdelle käyttäjälle

1 Johdanto

Tässä päättötyössä käsitellään pikaviestinten käyttöä yritysten viestinnässä. Työssä on erityisesti keskitytty pikaviestimistä IRC:n käyttöön, mutta osaa muistakin pikaviestimistä on sivuttu. Muita pikaviestimiä sivutaan lähinnä vertailtaessa IRC:n etuja ja haittoja niihin. Lisäksi työssä käsitellään IRC-bottien toimintaa ja hyödyllisyyttä viestinnän ja työnteon tehokkuuden lisäämiseksi.

Päättötyön tarkoituksena selvittää IRC:n mahdollisuuksia yritysten viestinnässä. Päättötyössä kerrotaan IRC:n eduista ja haitoista pikaviestimen käyttöönotossa. Työssä otetaan huomioon myös tietoturvan säilyttäminen viestinnässä IRC:n välityksellä. Lisäksi työssä kerrotaan IRC-bottien toiminnasta ja mahdollisuuksista. Päättötyön teon yhteydessä on tehty Eggdrop-tyyppinen botti sekä sille skripti.

2 IRC:n käyttö yritysten viestinnässä

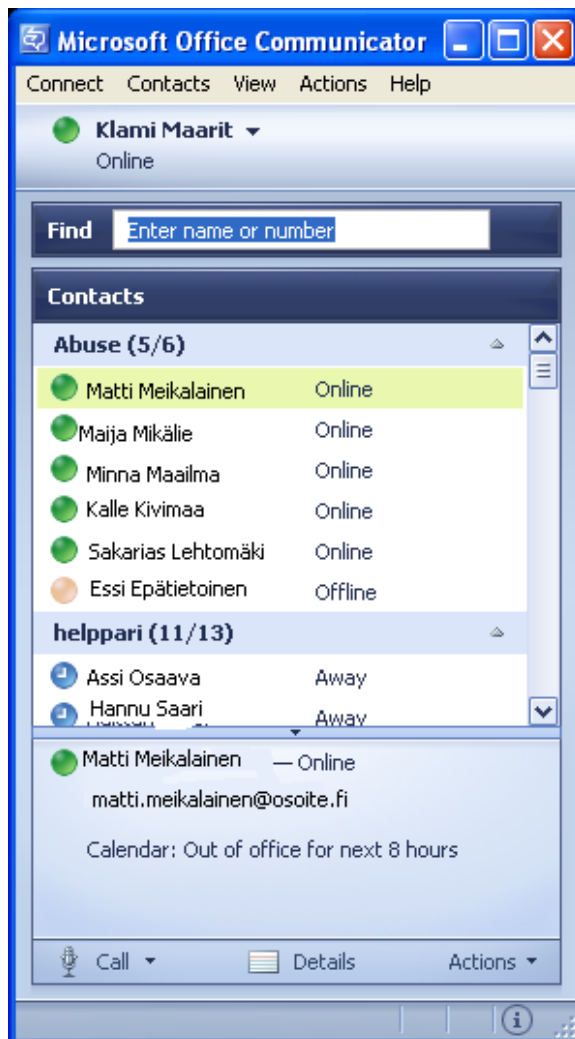
2.1 IRC ja Viestintä

Kommunikointikeinoja yrityksissä on monenlaisia. Suullista kommunikointia tapahtuu eri muodoissa, kahvipöytäkeskusteluista organisoituihin kokouksiin. Puhelimet ovat myös tärkeässä roolissa kommunikoinnissa, jos haluttu henkilö ei ole läheisyydessä. Nykyisin yrityksissä myös tietokoneen välityksellä tapahtuva kommunikointi on noussut merkittävään asemaan. Tähän tarkoitukseen on otettu käyttöön erilaisia pikaviestimiä sähköpostin lisäksi. Tavoitteena tälle kaikelle kommunikoinnille on, että tiedon kulku yrityksen sisällä ja sen ulkopuolelle olisi mahdollisimman tehokasta ja selkeää.

Internetin välityksellä keskustelua ja viestimistä voi tapahtua hyvin erilaisilla tavoilla. Yleisimmät viestimismuodot ovat sähköpostin välityksellä käytävä keskustelu, pikaviestinten avulla käytävä keskustelu sekä sosiaalisen median viestimismuodot. Sähköpostitse käytävä keskustelu eroaa pikaviestinten keskusteluista siten, että pikaviestimistä tiedetään, että toinen osapuoli on lähes varmasti vastaanottamassa viestin heti tai melko pian. Näin ollen myös oletetaan, että vastaanottaja pystyy

vastaamaan lähes reaaliaikaisesti viestiin. Sähköpostiviesteissä ei ole varmuutta siitä milloin vastaanottaja lukee viestin. Sosiaalisen median viestimismuodot ovat usein risteytyksiä sähköpostista sekä pikaviestimistä. Esimerkiksi Facebookissa ja Twitterissä viestejä voi jättää kaikkien nähtäväksi tilapäivityksin. Facebookin reaaliaikaisuus tulee ilmi esimerkiksi Facebookin keskusteluissa ja siinä, että Facebook näyttää paikalla olevia käyttäjiä.

Erilaisia yrityksissä käytettäviä pikaviestimiä ovat esimerkiksi IRC [3], Microsoft Office Communicator [25, s.109–113], Microsoft Messenger [25, s.19–21; 27, s.194–197], Skype [26, s.3], ICQ [27, s.190–194] sekä Facebookin keskustelu [23, s.76–77]. Näistä luetelluista Microsoft Office Communicator (MOC) on suunniteltu yritysten käyttöön. MOC:sta on esimerkki kuva kuviossa 1.



Kuvio 1. Microsoft Office Communicator.

Nykyään myös erilaiset sosiaaliset mediat toimivat yritysten kommunikointivälineenä, tosin lähinnä yrityksestä ulospäin suuntautuvassa viestinnässä. Esimerkiksi Facebook [23, s.3–5], Twitter [24, s.3–4] ja Google+ [22, s.16–19] ovat hyviä keinoja mainostaa yritystä ja olla läsnä asiakkaiden jokapäiväisessä elämässä. Yritysten sisäiseen kommunikointiin sosiaaliset mediat soveltuvat huonommin tietoturvariskien vuoksi. Esimerkiksi Facebook tallentaa kaiken Facebookiin kirjoitetun tiedon, jolloin yrityksen sisäiset tiedot päätyvät myös Facebookin tietoon [1]. Tässä opinnäytetyössä käsitellään tietokoneen välityksellä käytävistä pikaviestimistä erityisesti IRC:tä, joka on vanhin edelleen aktiivisessa käytössä oleva pikaviestin.

IRC:n sijasta yritykset voivat myös käyttää Facebookin keskustelun kaltaisia palveluita omille sivuilleen asennettuna. Näissä verkkokeskusteluissa yrityksen työntekijä vastaa reaaliajassa asiakkaiden esittämiin kysymyksiin. Keskustelut ovat yleensä vain yrityksen työntekijän ja asiakkaan välisiä. IRC:n etuna vastaavassa palvelussa on se, että useat käyttäjät näkevät samalla vastauksen kysymykseen ja samalla muidenkin ongelmat saattavat ratketa. Henkilökohtaisemmissa ongelmissa voi yleisen keskustelun sijasta kysyä ratkaisua ongelmaan yksityisessä keskustelussa.

2.2 IRC

2.2.1 IRC:n historia

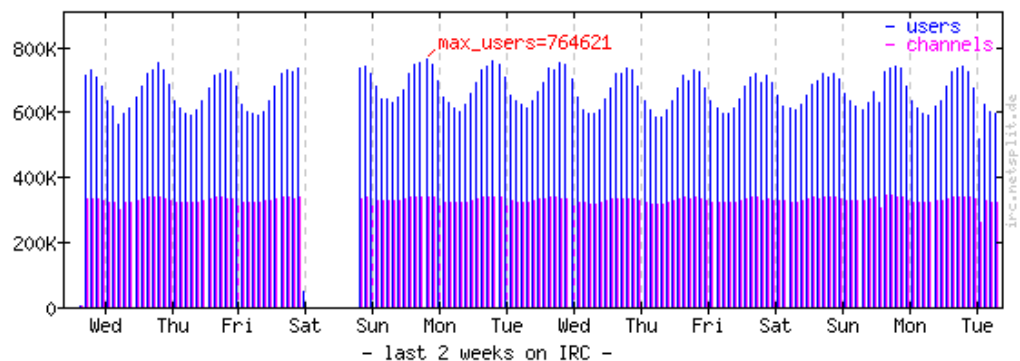
IRC on lyhennys sanoista Internet Relay Chat. IRC on pikaviestinpalvelu joka mahdollistaa reaaliaikaisen keskustelun internet-käyttäjien välillä. Se on avoin TCP/IP:n päällä toimiva sovellustason protokolla, jota käytetään ottamalla yhteys IRC-verkkoon asiakasohjelmalla [2]. IRC:n kehitti Jarkko Oikarinen vuonna 1988. IRC pohjautuu Jyrki Kuoppalan ohjelmaan MUT (MultiUser Talk). Aluksi IRC otettiin Oulun yliopistossa käyttöön elokuussa 1988, ja sillä oli vain noin 10 käyttäjää. Kuitenkin jo vuonna 1995 käyttäjiä saattoi olla pelkästään yhdessä IRC-verkossa (EFNet, joka oli silloin suurin IRC-verkko) yli 7000 samanaikaisesti. [3, s.11]. Nykyisin yhdessä IRC-verkossa (QuakeNET, joka on suurin IRC-verkko) voi olla jopa 70 000 käyttäjää. Yhteensä kaikissa IRC-verkoissa on noin 750 000 yhtäaikaista käyttäjää [4]. Kuviossa 2 on tarkemmin tietoa IRC-verkkojen yhtäaikaista käyttäjistä. Kahden viikon sisällä kuvion ottohetkestä suurin yhtäaikainen käyttäjämäärä IRC-verkoissa on ollut 764 621

käyttäjää. Kuvion ottohetkellä IRC-verkoissa oli 622 171 käyttäjää 327 692:lla eri kanavalla. Eri IRC-palvelimia puolestaan on ollut yhteensä 4645. Kanavien koko voi vaihdella yhdestä käyttäjästä useisiin satoihin käyttäjiin. Satunnaisten suurempien tapahtumien aikana, kuten WTC-iskun satuttua, voi aiheen mukainen kanava saada jopa tuhansia käyttäjiä.

Last measurement...

	known	reached	users	channels	servers
competitors:	650	647	545938	285392	4341
mavericks:	3	3	71855	40155	47
applicants:	52	52	4378	2145	257
total:	705	702	622171	327692	4645

Last 2 weeks...



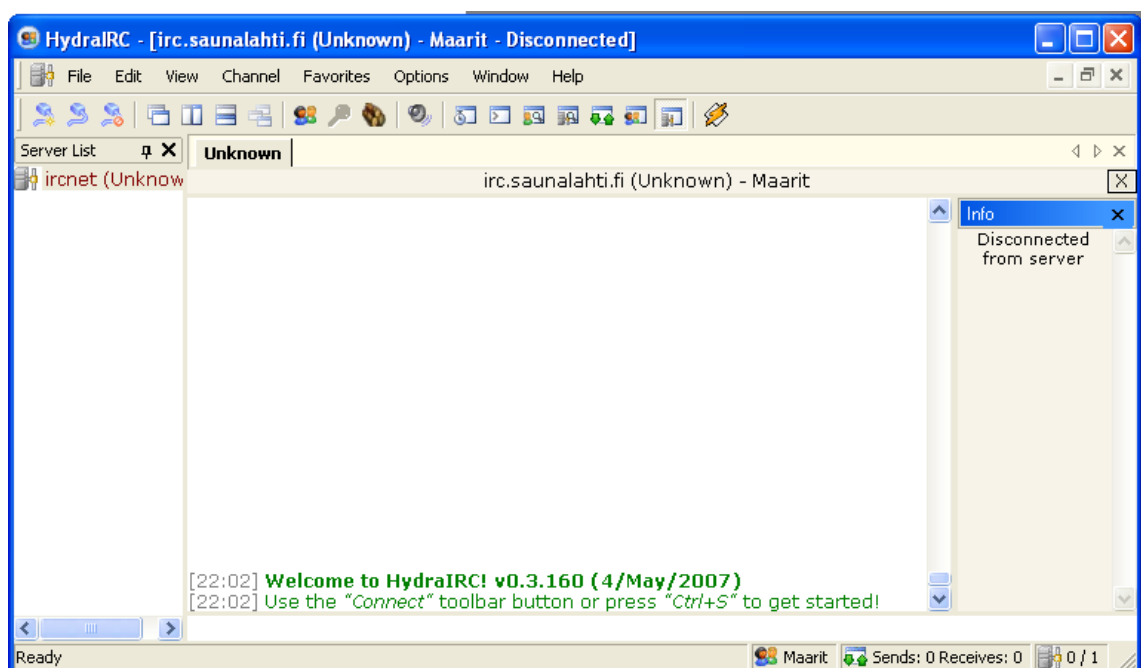
Kuvio 2. Tilastoja IRC:n käyttäjämääristä 20.3.2012 [4].

2.2.2 Asiakasohjelmat

IRC:n käyttöön tarvitaan asiakasohjelma, jonka takia IRC:n käytön aloittaminen ei ole niin helppoa kuin monien muiden Internetpalveluiden. Tämän seurauksena on kehitetty erilaisia selaimella toimivia chat-palveluita, joissa asiakasohjelman sijasta keskustelu toimii selaimen sisällä. Tosin nykyisin on olemassa myös selaimella käytettäviä palveluita, joilla pääsee IRC-verkkoihin. Kyseiset palvelut ovat usein epävakaita, ja käytettävyys on usein huonompi kuin IRC:lle tarkoitetuissa asiakasohjelmissa. Selaimella käytettävä palvelu on kuitenkin käytännöllinen silloin kun

asiakasohjelmaa ei saa asennettua koneelle tai haluaa käyttää IRC:tä ilman suurta opetettua. Yleinen selainpohjainen IRC-palvelu IRCnettiin on WebChat [30].

IRC:ssä käytettäviä asiakasohjelmia on saatavilla sekä ilmaisia (kuten HydraIRC) että maksullisia (kuten mIRC). mIRC [6] sekä HydraIRC [7] ovat graafisia asiakasohjelmia, jotka on tehty Windows-käyttöjärjestelmille. Useimmat aloittavat IRC:n käytön graafisilla asiakasohjelmilla, koska niiden käyttö mielletään helpommaksi. Kuviossa 3 on esimerkkikuva graafisesta asiakasohjelmasta. Unix-järjestelmillä on omat asiakasohjelmansa, esimerkiksi Irssi [8]. Irssi on tekstipohjainen asiakasohjelma, minkä takia käytön aloituskynnys on hieman korkeampi kuin graafisissa asiakasohjelmissa. Tekstipohjaissa asiakasohjelmissa on tiedettävä ja kirjoitettava komennot, kun puolestaan graafisissa asiakasohjelmissa tarvittavat toiminnot saa suoritettua hiiren painalluksella. Irssiä pystyy käyttämään myös Windows-koneilla käyttämällä esimerkiksi Putty asiakasohjelmaa [9], jolla otetaan etäyhteys palvelimeen shell-tunnusten avulla. Shell-tunnuksilla tarkoitetaan tunnuksia palvelimeen, jonka avulla käyttäjät voivat olla etänä yhteydessä jatkuvasti päällä olevaan koneeseen mistä tahansa.



Kuvio 3. HydraIRC on graafinen asiakasohjelma IRC:lle.

2.2.3 IRC:n keskustelumuodot

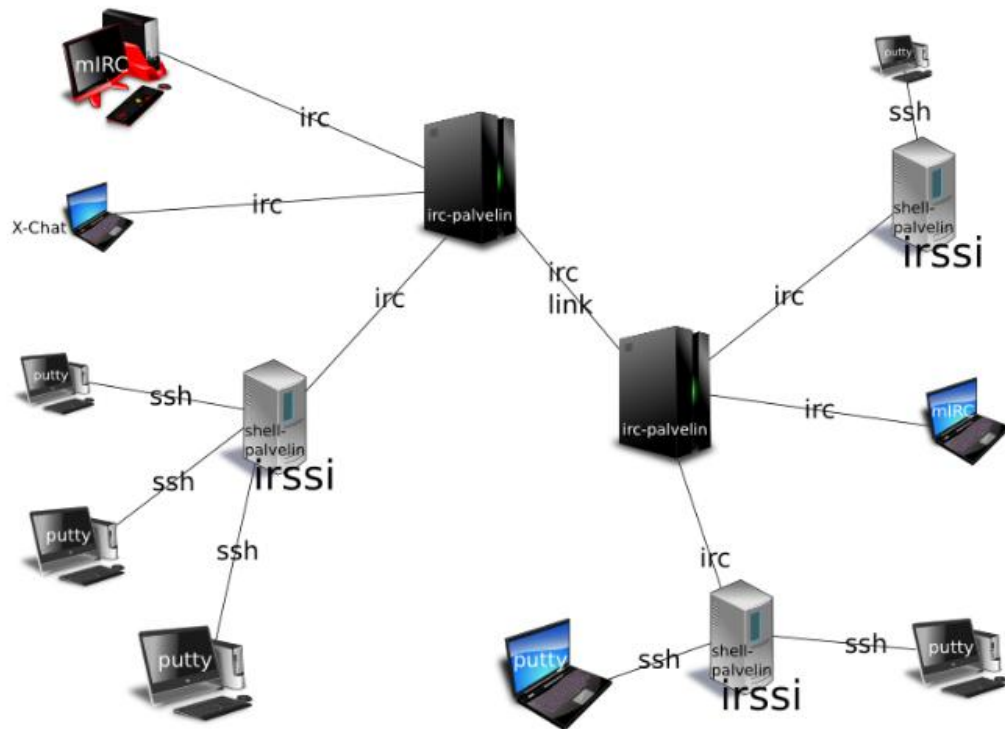
IRC:ssä voi keskustella yksityisenä keskusteluna haluamansa käyttäjän kanssa tai vaihtoisesti voi luoda kanavan, jossa keskustelijoita voi olla useampia. Kanavamuotoinen keskustelu on IRC:n pääasiallinen keskustelumuoto. Kanavat voivat olla joko kaikille suunnattuja kanavia kuten, esimerkiksi IRCnetin kanava #freenet, johon IRC:n alkuvaiheessa useat käyttäjät automaattisesti liittyvät. Kanava voi myös olla jonkin harrastuksen tai asian kanava, johon kaikki asiasta kiinnostuneet voivat liittyä. Esimerkki tällaisesta kanavasta on #linux, joka on siis suunnattu Linux-käyttöjärjestelmistä kiinnostuneille. Kanavat voivat olla myös yksityisiä, jolloin kanavan voi suojata salasanalla. Tämän jälkeen vain salasanan tietävät pääsevät kanavalle [3, s.152; 2]. Kanavan käyttäjiä voi myös rajoittaa siten, että kanavalle pääsee ainoastaan kutsuttuna, jolloin kanavalla ei ole salasanaa mutta ainoastaan jo kanavalla olevat henkilöt voivat kutsua kanavalle uusia henkilöitä.

2.2.4 IRC-verkon toiminta

Kuviossa 4 esitellään IRC-verkon toiminta. Eri IRC-verkot eivät ole toisiinsa yhteydessä mutta saman IRC-verkon IRC-palvelimet ovat keskenään yhteydessä. Kuviossa 4 kahden IRC-palvelimen välillä on vedetty "IRC link"-viiva. Palvelimia voi olla useita samassa IRC-verkossa. Palvelimelta voi puolestaan mennä suoraan käyttäjän omalle koneelle "IRC"-viiva, jolloin viivan päästä löytyy käyttäjän käyttämä asiakasohjelma kuten mIRC. Palvelimelta voi myös mennä yhteys shell-palvelimelle, jossa pyörii esimerkiksi Irssi, johon shell-palvelimen asiakkaat voivat sitten ottaa yhteyden omalta koneelta ssh:n avulla esimerkiksi Putty-asiakasohjelmalla.

IRC-palvelimet ovat keskenään yhteydessä, siten että jokainen palvelin on yhteydessä jokaiseen palvelimeen. Jos kahden palvelimen välille tulee katkos, niin näillä palvelimilla olevat käyttäjät eivät näe toistensa tekstejä, mutta samalla palvelimella tai niillä palvelimilla, jotka eivät ole menettäneet yhteyttään, ovat käyttäjät näkevät toisensa. Yritysten sisäisissä IRC-verkoissa kannattaa käyttää vain yhtä palvelinta. Yleensä käyttäjämäärät yrityksen sisällä ovat niin pieniä, että yksi palvelin riittää. Vain yhden palvelimen ollessa käytössä ei katkoksia palvelimien välille pääse syntymään. IRC-verkossa asiakasohjelman lähettäessä viestin palvelin vastaanottaa sen ja välittää

eteenpäin muille palvelimille. Sitten palvelimet lähettävät viestin eteenpäin käyttäjille, jotka ovat liittyneet kanavalle, jolle viesti osoitettiin. Yksityisviestit välittyvät samaan tapaan, mutta vain viestin kohteelle itselleen.



Kuvio 4. IRC-verkon toteutus. [5.]

Shell-palvelun käyttämisen etuna on se, että palvelin on pääsääntöisesti käytössä koko ajan, jonka takia käyttäjä voi sammuttaa oman koneensa ilman yhteyden katkeamista palvelimen ja asiakasohjelman välillä. Tällöin on mahdollista lukea IRC:ssä tapahtuneet asiat jälkeenpäin käyttäjän palatessa takaisin koneelle.

2.2.5 Nimimerkkien käyttö

IRC:ssä on tapana käyttää nimimerkkejä, jotka eivät välttämättä liity oikeaan nimeen. Omaakin nimeä voi käyttää nimimerkkinä mutta rajoituksena on, että sama nimimerkki voi olla vain yhdellä käyttäjällä kerrallaan. Lisäksi IRC-verkossa voi olla rajoituksia nimimerkin pituuden ja merkistön suhteen. Esimerkiksi IRCnetissä oli ennen rajoituksena 9 merkkiä mutta pituus on kasvatettu nyt 15 merkkiin [3, s.56; 2]. Lisäksi

rajoituksena on, että nimessä ei voi käyttää skandinaavisia merkkejä ja vain osaa erikoismerkeistä [3, s.56; 2]. Nimimerkkiä ei periaatteessa voi varata itselleen, minkä seurauksena oman nimimerkin voi menettää toiselle tai toinen voi esiintyä toisena käyttäjänä käyttämällä varastettua nimimerkkiä. Käytännössä kuitenkin uusi käyttäjä pyrkii löytämään nimimerkin joka ei ole käytössä toisella käyttäjällä. Nimimerkin saa pidettyä itsellään esimerkiksi käyttämällä shell-palveluita, jolloin käyttäjä on koko ajan verkossa ja nimimerkki pysyy siten varattuna. Nimimerkki on mahdollista varata itselleen käyttämällä NickServ-nimistä palvelua, joka löytyy osasta julkisia IRC-verkkoja kuten Efnetistä. Nickserv-palvelussa nimimerkki varataan itselleen sähköpostin ja salasanan avulla. [29].

2.2.6 Käyttäjän tiedot

Oikean nimen voi kuitenkin asettaa näkyväksi, jolloin sen voi katsoa "whois"-komennon avulla [2]. Useilla kanavilla myös vaaditaan oikean nimen näkymistä. Kuviossa 5 on esimerkki "whois"-komennolla saatavista tiedoista. Ensin on vaaleammalla värillä nimimerkki, jonka perässä on käyttäjän identti, joka on käyttäjän yksilöllinen tunniste sekä isäntänimi, joka on tietokoneen nimi tai IP-osoite. Tämän rivin alta löytyy käyttäjän asettama nimi, joka voi olla oikea nimi tai mitä tahansa käyttäjän asettamaa tekstiä. Nimen alta löytyvät käyttäjän kanavat. Tässä ei välttämättä näy yhtään kanavaa, jos käyttäjä ei ole millään kanavalla tai vaihtoehtoisesti vain kanavilla, jotka on asetettu salaisiksi. Kanavien alta löytyy käyttäjän käyttämä IRC-palvelin. Tämän alapuolelta löytyy puolestaan poissaoloviesti, jos katsottu käyttäjä on sellaisen asettanut. Lisäksi Irssi tarjoaa myös "whois"-komennolla tiedon ajasta, jonka käyttäjä on ollut tuottamatta tekstiä IRC-verkkoon. mIRC:llä kyseisen ajan saa kirjoittamalla komennon "whois nick nick".

```
22:10 -!- End of WHOIS
22:14 -!- Tassukka [tazze@hauki.tunkki.fi]
22:14 -!- ircname : Maarit Klami
22:14 -!- channels : #inssiharjoitus
22:14 -!- server : irc.elisa.fi [Elisa Oyj, Finland]
22:14 -!- away : poissa
22:14 -!- idle : 0 days 0 hours 15 mins 18 secs [signon: Tue Mar 20
00:09:59 2012]
22:14 -!- End of WHOIS
```

Kuvio 5. "Whois"-komennolla saadaan seuraavat tiedot.

2.3 Hyödyt IRC:n käytössä yritysten viestinnässä

IRC on monipuolinen viestintäväline. Se toimii reaaliaikaisena tai sen välityksellä voi jättää viestejä myöhemmin luettavaksi. Tämä mahdollistaa poissaolon aikana käytyjen keskustelujen lukemisen myöhemmin käyttäjälle sopivana ajankohtana. Tällöin käyttäjä voi tarkistaa, onko poissaolon aikana tapahtunut jotain, mikä vaatii huomiota tai toimenpiteitä. IRC:n hyödyt korostuvat tilanteessa, joissa useampi henkilö tekee samaa työtä. Tulevan tehtäväpyynnön voi helposti kuitata IRC:ssä itselleen, muussa tapauksessa se pitäisi esimerkiksi lähettää sähköpostilla kaikille. Tästä aiheutuisi turhaa sähköpostiliikennettä niille, jotka eivät kyseistä tehtäväpyyntöä tällä kertaa käsittele.

IRC:n etuna muihin pikaviestimiin verrattuna on, että IRC:ssä useiden ihmisten kanssa keskusteleminen on tehty sulavaksi, mutta yksityisetkin keskustelut onnistuvat. IRC:ssä pystyy luomaan kanavia erilaisille toiminnoille. Esimerkiksi teleoperaattorilla voisi olla seuraavanlaisia kanavia, kuten #tiedotus, jossa olisi kaikkia koskevat tärkeät ilmoitukset. Toisena kanavana voisi olla #mobiilit, joka on kanava, jossa olisi mobiilipuolen osajia. Pienemmälle ryhmälle kanavan nimenä voisi olla #Mobiilitiimi2, joka olisi ryhmän oma sisäinen kanava. Näiden lisäksi vielä kanava #sekalaista, joka olisi kanava, jossa kaikki voivat jutella vapaasti. Kanavien nimen edessä käytetään usein #-merkkiä ilmaisemaan, että kyseessä on kanava.

Joissain IRC-verkoissa kanavan edessä voi olla myös &-merkki tai !-merkki, jotka muuttavat hieman kanavan ominaisuuksia. Mikäli kanavan nimen edessä on &-merkki, kanavalle pääsee vai tietyn IRC-palvelimen kautta. !-merkillä kanavan nimen edessä saadaan kanavan luomisen yhteydessä kanavalle uniikki nimi, esimerkiksi !ueat7FirmanNimi. Tämä liittyy palvelinten välisten yhteyksien katkeamisista johtuvien ongelmien minimoimiseen [10]. Jos yhteydet katkeaa palvelinten välillä, voi seurauksena olla, että sama kanava on kahteen kertaan verkossa eri palvelimilla, jolloin väärät henkilöt voivat saada operaattorin oikeudet, ja siten kanava lopulta päätyä vääriin käsiin. Kanavia voi luoda suuriakin määriä erilaisille asioille ja samanaikaisesti käyttäjä voi olla useammalla eri kanavalla. Lisäksi jos haluaa keskustella vain tietyn käyttäjän kanssa, voi hänelle muodostaa oman yksityisen keskustelun.

Kaikki käydyt keskustelut voi myös kerätä talteen niin kutsutuiksi lokeiksi [3, s.53]. Näin voi myös jälkeenpäin tarkastaa, mitä kanavalla tai yksityisessä keskustelussa on

puhuttu. Keskusteluista pystyy myös korostamaan tietyt sanat. Korostaminen toimii siten, että se esimerkiksi muuttaa halutun sanan tai halutun sanan sisältävän lauseen väriä muusta tekstistä erottuvaksi. Lisäksi korostus voidaan asettaa myös tuottamaan äänimerkin. Tämä auttaa huomaamaan, jos keskustelu koskee käyttäjää tai käyttäjää kiinnostavaa aihetta. Tällöin ei tarvitse välttämättä lukea kaikkea tekstiä kanavalta, kun itselleen merkitykselliset tekstit näkyvät korostettuna. Esimerkiksi, jos työtehtäviin kuuluvat tietoturva-asiat, korostamalla sanan tietoturva IRC ilmoittaa aina kun kyseinen sana mainitaan kanavalla. Koneelta poistuessaan voi merkitä itsensä poissaolevaksi. Kun itsensä on laittanut poissaolevaksi, viestit, joissa on mainittu korostetut sanat, jäävät poissaololokiin, josta ne palatessaan näkee helposti. Kuviossa 6 on esimerkki Tazze- nimimerkkisen käyttäjän poissaololokista. Poissaoloviestillä pystyy myös kertomaan muille käyttäjille tietoa poissaolon syystä ja kestosta.

```
22:17 #lontoo: < Tassukka> Tazze: Moi! kerkeetkö huomenna tulemaan palaveriin?
22:18 #lontoo: < Tassukka> Tazze: Unohdin ostaa kissanruokaa. Tuotko kun
                samalla kun tulet kotia
22:19 <Tassukka> Tuletko huomenna syömään mun kanssa lounaalla?
22:20 #lontoo: < Tassukka> Oletteko nähnyt Tazzea? pitäisi kysyä yhtä juttua
```

Kuvio 6. Keltaisella olevat nimimerkit ovat kanavalta tulleita poissaoloviestejä ja punaisella nimimerkillä oleva viesti on tullut yksityisestä keskustelusta.

Paikalla ollessaankin kaiken tekstin kanavilta tai yksityisistä keskusteluista voi kerätä erilliseen lokitiedostoon, josta käytyjä keskusteluja voi tarkastella jälkikäteen. Normaalisti asiakasohjelmat näyttävät vain lyhyen historian aikaisemmista keskusteluista. Lokitiedoston voi myös rajata yksittäisille kanaville tai vain pelkästään yksityisiin keskusteluihin.

Tietyn käyttäjän voi tavoittaa helposti tietämällä hänen nimimerkinsä. Tällöin ei tarvitse odottaa uuden keskustelukumppanin hyväksyntää keskustelun aloitukseen, kuten esimerkiksi Microsoft Office Communicatorissa tehdään. Jos kaikki määrittelevät oikean nimen tietoihinsa, on mahdollista nähdä kanavalla olevien käyttäjien oikeat nimet ja sitä kautta tavoittaa haluamansa käyttäjän. Nimimerkit voisivat olla esimerkiksi työpaikan osoitekirjassa näkyvissä, mikä helpottaisi nimimerkin ja käyttäjän yhdistämistä.

Julkisessa IRC-verkossa oleva kanava voi toimia myös yrityksen yhtenä asiakasrajapintana. Asiakas voi tulla yrityksen kanavalle, jossa on paikalla henkilö tai useita henkilöitä yrityksestä paikalla kertomassa ratkaisuja ongelmiin tai vastaamassa muuten kysymyksiin. Kanavaa voidaan pitää myös asiakkaille tiedottamisen yhtenä väylänä, jossa kerrotaan, jos yrityksellä on esimerkiksi palvelutason alenemaa. IRC tarjoaa helpon tavan jakaa keskusteluyhteys useamman henkilön kanssa, joiden ei tarvitse edeltä käsin tuntea toisiaan.

2.4 IRC:n heikkoudet muihin pikaviestimiin nähden

IRC:n huonona puolena on hankalampi käyttöönotettavuus muihin pikaviestimiin verrattuna. Useimmat muut pikaviestimet ovat muutamalla hiiren painalluksella asennettavissa ja käyttövalmiina. IRC:ssä sen sijaan tarvitsee olla sopiva asiakasohjelma ja tietää monista eri IRC-verkoista, mihin niistä pitäisi kytkeytyä. Tunnetuimpia ja suurimpia IRC-verkkoja on IRCnet, EFNet, Quakenet sekä UnderNet [11 s.106–107]. Yrityksillä voi olla myös omia IRC-verkkoja, joihin ulkopuoliset eivät pääse kytkeytymään. Verkon valinnan lisäksi täytyy vielä tietää, minkä palvelimen kautta haluamaansa IRC-verkkoon pääsee kytkeytymään. Jos yrityksellä on oma IRC-verkko, on sillä yleensä myös oma palvelin, johon tulee kytkeytyä. Käytettäessä julkisia IRC-verkkoja voi sopivan palvelimen löytäminen olla haastavaa niissä olevien käyttäjämäärärajoitusten vuoksi. Monilla oppilaitoksilla on tarjolla palvelimia, joiden välityksellä myös muut kuin kyseisen oppilaitoksen opiskelijat, pääsevät verkkoon. Esimerkkinä oppilaitoksen palvelimesta on Aalto-yliopiston IRCnet-palvelin irc.cc.hut.fi. Verkon valinnan jälkeen tarvitsee vielä tietää sen kanavan nimi, jossa halutut henkilöt ovat tai jossa kiinnostavia asioita käsitellään. IRC:tä voi käyttää myös pelkästään yksityisiin keskusteluihin, joskin silloin IRC:n hyödyt jäävät pieniksi. IRC:n monimutkaisuutta lisäävät erilaiset käyttömahdollisuudet, vaikka jo hyvin pienellä määrällä osaamista IRC:tä pystyy käyttämään riittävän tehokkaasti.

Oikealla nimellä käyttäjän löytäminen IRC:stä on huomattavasti hankalampaa kuin esimerkiksi Microsoft Office Communicatorilla (MOC). MOC:ssa on hakemisto, josta voi hakea henkilöitä oikealla nimellä, tosin vain yrityksen sisältä. IRC:ssä helpoin tapa löytää käyttäjä on tietää valmiiksi etsittävän käyttäjän nimimerkki. Nimimerkit voivat olla käyttäjän nimestä johdettuja, jolloin oikean nimimerkin päättelemisen on

helpompaa. Esimerkiksi jos Maija Meikäläinen olisi mobiiliosaaaja, voisi hänen nimimerkinsä olla MaijaMeMob. Nimimerkit voidaan vakioda esimerkiksi siten, että ne muodostetaan etunimestä ja sukunimen kahdesta ensimmäisestä kirjaimesta sekä työtehtävästä. Nimimerkkien ongelma on myös se, ettei niitä voi aina varata itselleen, minkä takia nimimerkin tietäminen voi olla hankalaa, jos tavoitettava henkilö on joutunut tai muuten halunnut vaihtaa nimimerkkiään. Muissa pikaviestimissä nimimerkki varataan itselleen rekistöityessä ja siten se säilyy samana aina. Tämä helpottaa halutun käyttäjän löytymistä.

Heikkoudeksi IRC:n käytössä voidaan laskea myös, että keskustelu saattaa harhautua aiheesta helposti. Näin voi tietenkin tapahtua myös kasvotusten käydyssä keskustelussa, etenkin kun keskustelijoita on useita. Tähän usein ratkaisuna on kanava, missä aihe on vapaa ja keskustelun siirtyessä asiasta pois voi kanavan operaattori tai muut keskustelijat ohjata aiheen vierestä keskustelijat vapaa-aiheiselle kanavalle. Voi kuitenkin olla, ettei kukaan huomaa siirtää keskustelua toisaalle ja jälkikäteen kanavaa lukevat saattavat turhautua aiheen vierestä käydyn keskustelun määrään.

Ongelmia voi seurata myös tilanteessa, kun on paljon samanaikaisia keskustelijoita ja tekstiä tulee nopealla tahdilla. Tällöin keskustelu vaatii kaiken huomion, vaikka välttämättä kaikki sillä hetkellä kanavalle tuleva ei olisikaan merkityksellistä kaikille lukijoille. Jos keskustelun aikana joutuu poistumaan IRC:stä voi osassa asiakasohjelmia käyttää palvelua, jossa tulee merkintä keskusteluun siihen kohtaa, kun käyttäjä siirtyy kanavalta toiselle. Takaisin IRC:hen tai kanavalle tullessa on helppo selata tekstiä taaksepäin ja nähdä, missä kohtaa keskustelua on poistunut siitä.

2.5 Erot pienissä ja suurissa yrityksissä

Pienissä yrityksissä IRC:n käyttöönotto voi olla helpompaa kuin suurissa yrityksissä. Yrityksen henkilökunnan määrän ollessa pienempi on palvelun käyttöön opettavien käyttäjien määräkin pienempi. Lisäksi usein kaikki tuntevat toisensa ja tunnistaminen IRC:ssä helpottuu. Hankaluutta aiheutuu lähinnä, että pienelle yritykselle ei välttämättä ole omaa IRC-verkkoa, mistä saattaa seurata ongelmia tietoturvamielessä. Suurissa yrityksissä suurin ongelma IRC:n käytössä on sama kuin muissakin pikaviestimissä;

kuinka saada kaikki käyttämään samaa tai edes jotain pikaviestintä. Isoissa yrityksissä on paljon erilaisia ja eritasoisia osajia, minkä takia uuden työkalun oppiminen voi kestää pidempään kuin pienemmässä yrityksessä. Isommissa yrityksissä on todennäköisemmin myös henkilöitä, jotka ovat tehneet asiat aina samalla tavalla ja halua toimintatapojen muuttamiseen ei löydy.

Pikaviestinten edut kuitenkin toteutuvat isoissa sekä pienissä yrityksissä yhtä hyvin. Tosin suuressa yrityksessä IRC:n edut kanavamuotoisessa keskustelussa ovat huomattavat muihin pikaviestimiin verrattuna. Usein samaa asiaa tekee useampi henkilö, joten kanavalla asiasta sanoessa kaikki osajat huomaavat eikä jokaiselle tarvitse sanoa asiasta erikseen. Ei myöskään tarvitse tietää, ketkä osajista ovat töissä milloinkin, kun IRC:ssä jokainen töissä oleva osaja voi lukea kerrotun asian. Näin esimerkiksi lomallaolijat välttyvät lukemasta sähköposteja aiheista, jotka ovat jo ratkenneet.

3 Tietoturva IRC:n käytössä yrityksen viestinnässä

3.1 Tietoturva

Tietoturva on erittäin oleellinen asia kaikessa yrityksen viestinnässä, olipa se suullista, sähköpostitse tai pikaviestitse käytävää. Salaustekniikoihin ja tekniseen toteutukseen kannattaa panostaa riittävästi, mutta tärkeintä on selvittää henkilöstölle tietoturvariskit, ja tiedot mitkä kuuluvat vain yrityksen sisälle, tai esimerkiksi vain ryhmän sisäiseen tietoon [32, s.25–26, 113–114]. Viestintäviraston tietoturva-asiantuntija Antti Kiurun kommentin mukaisesti ”vaarallisin uhka olet sinä itse” [12]. IRC:ssä tietoturva pohjautuu lähinnä käyttäjien valppauteen ja SSL-salausprotokollan käyttöön. Turvallisuutta voidaan lisätä rajaamalla käyttäjiksi vain yrityksen työntekijät. Lisäksi salaamalla kanava joko salasanalla tai laittamalla kanava sellaiseksi että sinne pääsee vain kutsuttuna, rajaa tuntemattomat käyttäjät pois kanavalta. Tietoturvaa voi lisätä käyttämällä IRC-botteja, jotka voivat vartoida kanavaa, ettei se joudu vieraiden käyttäjien haltuun tai että kanavalle ei pääse epätoivotut käyttäjät.

3.2 SSL-salausprotokolla

Osassa palvelimista on mahdollista nykyisin käyttää SSL-salausprotokollaa, Secure Socket Layer. Tämän ansiosta julkisessakin IRC-verkossa keskusteluista tulee turvallisempia, kun liikenne kulkee palvelimelle salattuna. SSL-salausprotokollalla voidaan suojata internet-sovellusten kuten IRC:n tietoliikenne IP-verkossa. SSL-salausprotokollan varmennus perustuu sertifikaattiin, jonka palvelin lähettää asiakasohjelmalle [13].

IRCnetissä ja QuakeNetissä ei ole kummassakaan tällä hetkellä mahdollisuutta käyttää SSL-salausprotokollaa. EFnetissä, joka on maailmanlaajuisesti käytetyin IRC-verkko, on mahdollisuus SSL-salausprotokollan käyttöön [14]. IRC-verkon voi myös tehdä sellaiseksi, että kaikkien palvelinten on pakko käyttää SSL-salausprotokollaa, tai kanavalle voi laittaa asetuksen, joka estää pääsyn muilta kuin SSL-salausprotokollan käyttäjiltä. SSL-salausprotokollan käyttöön pakottaminen lisää turvallisuutta, koska tällöin kukaan ei käytä salaamatonta yhteyttä. Jos yhdelläkin kanavan käyttäjistä on salaamaton yhteys, kaikkien tietoturvasuus heikkenee.

3.3 Sosiaaliset uhat

Sosiaalisilla uhilla tarkoitetaan tilannetta, jossa henkilökuntaan kuuluva henkilö paljastaa jotain yrityksen sisäistä tietoa tai muuten salassa pidettävää tietoa jollekin ulkopuoliselle tai yrityksen työntekijälle, joille tiedot eivät kuulu. Tiedon paljastaminen väärille henkilöille ei välttämättä ole tarkoituksellista, esimerkiksi IRC:ssä saattaa vahingossa liittää leikepöydältä tekstiä keskusteluun, johon sitä ei ollut tarkoitettu. Riskiä saa pienennettyä asettamalla asiakasohjelma varmistamaan ennen jokaista tekstin liittämistä, haluaako käyttäjä varmasti lähettää tekstin kanavalle. On myös mahdollista rajata, montako riviä tekstiä saa liittää ilman varmistamista. Kuviossa 7 esitellään tilanne, jossa käyttäjä liittää asiakasohjelmaan useamman rivin tekstiä ja asiakasohjelma varmistaa luvan, saako tekstin liittää. [32, s.25–26, 113–114].

```

22:27 < Tassukka> ... tarkoitettu. Riskiä saa pienennettyä asettamalla
      asiakasohjelma kysymään ennen jokaista tekstin liittämistä,
      oletko varma että haluat lähettää tekstin kanavalle. On myös
      mahdollista rajata montako riviä tekstiä saa liittää ilman
      varmistamista.
22:32 -!- Irssi: Pasting 1 lines to #inssiharjoitus. Press Ctrl-K if you wish
      to do this or Ctrl-C to cancel.
[22:32] [Tassukka(+i)] [55:sa [Act: 3,4,5,10,11,12,13,14,15,16,20,22,23,26,28]
[#inssiharjoitus] Hit Ctrl-K to paste, Ctrl-C to abort?

```

Kuvio 7. Kuvassa käyttäjä yrittää liittää IRC:hen yli yhden rivin pituisen viestin ja asiakasohjelma varmistaa, haluaako käyttäjä varmasti tehdä tämän. Painamalla Ctrl-K viestin saa lähettää kanavalle. Puolestaan Ctrl-C:tä painamalla ei viestiä lähetetä.

Tietoa saatetaan myös kertoa eteenpäin, jos käyttäjä ei tiedä informaation olevan luottamuksellista. Käyttäjä voi myös olettaa, ettei tiedon edelleen kertomisesta seuraa haittaa. Vahinko voi myös käydä, jos kirjoittaa tekstiä IRC:ssä väärälle kanavalle, tai jos kanavalle on saapunut henkilöitä, jotka sinne eivät kuulu. Ongelmana on myös, ettei oikeasti voi tietää, ketkä tekstiä lukevat. Kanavalla olijoista joku on voinut jättää tietokoneensa lukitsematta, ja ulkopuolinen henkilö lukee tekstiä tekstin muiden tietämättä. IRC:ssä on myös mahdollista kaapata toisen ihmisen nimimerkki ja sitä kautta esiintyä toisena henkilönä. Muut käyttäjät voivat kuitenkin havaita nimimerkin kaappaamisen tarkastamalla käyttäjän identin ja isäntänimen. Osalla IRC-palvelimia pystyy nimimerkin varaamaan yhdelle käyttäjälle, mikä lisää turvallisuutta. Erityisesti on tärkeää kouluttaa yritysten työntekijöille perusteet tietoturvasta ja yrityksen luottamuksellisten tietojen käsittelystä.

3.4 Viestintä yrityksen sisäsällä ja yrityksen ulkopuolella

Tietoturvan kannalta tärkeä päätös on, halutaanko IRC-verkko pitää yrityksen sisäisenä vai pääsevätkö myös yrityksen ulkopuoliset käyttäjät verkkoon käsiksi. Tietoturva on yrityksen sisäisessä IRC-verkossa huomattavasti parempi. Tämän takia etenkin isoimmissa yrityksissä, joissa IRC:tä käytetään työasioiden käsittelyyn, kannattaa käyttää omaa verkkoa. Lisäksi omassa verkossa asetukset pystytään laittamaan yrityksen tarpeiden mukaisiksi.

Yrityksen ulkopuoliset käyttäjät mahdollistava IRC-verkko avaa taas täysin uusia ovia yrityksen viestintään ja asiakaspalveluun. Päästämällä asiakkaat samaan verkkoon työntekijöiden kanssa pystytään esimerkiksi asiakaspalvelun väyliä kasvattamaan.

Puhelimen lisäksi asiakkaalla on mahdollisuus ottaa yhteyttä IRC:ssä yrityksen asiakas- tai vikapalveluun. Jos esimerkiksi vikapalvelulla on oma kanava IRC:ssä ja asiakas kysyy kanavalla ongelmasta, voivat useat asiakkaat saada tiedon häiriöstä kerralla, eikä kaikkien tarvitse välttämättä soittaa vikapalveluun erikseen. Kanavalle pystyy myös asettamaan kokoaikaisesti näkyväksi kanavan otsikon, jossa voidaan kertoa esimerkiksi vikatilanteista. Kanavalle voi myös automaattisesti päivittää reaaliaikaisen tilanteen tiedossa olevista heikentymistä palveluissa.

Välttämättä ei tarvitse päättää, käytetäänkö IRC:tä vain yrityksen sisällä vai ulkopuolella. Yrityksen kanavat voidaan jakaa tarpeen mukaan eri IRC-verkkoihin. Yrityksen sisäiset tiedot käsitellään kanavilla, jotka ovat yrityksen sisäisessä verkossa, kun taas asiakkaille suunnatut kanavat löytyvät julkisten IRC-verkkojen puolelta. Tämä tietysti edellyttää enemmän osaamista, kun käyttäjän pitää tietää, miten eri IRC-verkkoihin liitytään samanaikaisesti. Vaihtoehtoisesti käyttäjällä voi olla useita asiakasohjelmia auki siten, että eri IRC-verkot ovat eri asiakasohjelmissa. Jälkimmäinen käytäntö lisää myös osaltaan tietoturvaa. Jos yrityksen sisäiset kanavat ovat omassa ohjelmassaan ja julkisen puolen kanavat ovat toisessa ohjelmassa, vähenee riski kirjoittaa vahingossa väärälle kanavalle tekstiä.

4 IRC-botti

4.1 IRC-bottien käyttö

Botti eli IRC-bot on tietokoneohjelma tai skripti, joka näyttää muille käyttäjille tavalliselta IRC-käyttäjältä, joka suorittaa automaattisesti esimerkiksi kanavan hallintaa. Bottia komentaa botin tekijä tai käyttäjä, jonka botin tekijä on valttuuttanut botin haltijaksi. Botin voi myös asettaa tuntemaan tiettyjä käyttäjiä, jotka pystyvät salasanan avulla komentamaan bottia tietyissä rajoissa. Suurimmalle osalle IRC-käyttäjistä botit ovat lähinnä vain viihdykettä, mutta botteja pystyy myös käyttämään hyödyksi. Bottien avulla pystyy tekemään erilaisia asioita ja vain mielikuvitus on rajana sille, mitä ominaisuuksia botille voi toteuttaa.

Muita nimityksiä botille on robottiohjelma tai bottiohjelma. Laajemmin kuin vain IRC:hen liittyen botilla tarkoitetaan ohjelmaa, joka toimii itsenäisesti sille määriteltyjen toimintaohjeiden mukaan. Botteja käytetään etenkin silloin, kun tarvitaan suurta työmäärää, toistoa tai jatkuvaa päivystystä. Muita botteja IRC-botin lisäksi on esimerkiksi hakuagentti.[28].

Botti vastaanottaa palvelimelta kaikki viestit kanavilta, joille se on liitetty. Kanavalle tulleesta tekstistä botti etsii sanoja tai asioita, joita siihen on asetettu seurattavaksi. Yleisimmin botti toimii, kun sille annetaan jokin määrätty komento, kuten !sää. Usein boteille osoitetut komennot alkavat !-merkillä. Botin voi laittaa myös reagoimaan erilaisiin tilanteisiin. Esimerkiksi jos käyttäjä kirjoittaa kanavalle "Ketä se oli?", botilta voi tulla kommentti "Oikeasti se kirjoitetaan "Kuka hän oli?". Botti voi myös tuottaa kanavalle tekstiä esimerkiksi, jos botin nimimerkki kirjoitetaan kanavalle. Tällä ominaisuudella saadaan hämättyä muita käyttäjiä luulemaan bottia oikeaksi henkilöksi. Botin voi laittaa myös keräämään tietoa kanavalta. Esimerkiksi se voi listata halutulle internet-sivulle kaikki internetosoitteet, joita käyttäjät kanavalle lähettävät, jolloin on helppo jälkeenpäin tarkastaa, mitä sivuja kanavalla on käsitelty. Se voi myös reagoida käyttäjien lähettämiin internetosoitteisiin siten, että kertoo kyseisen sivun otsikon kanavalle. Tämä auttaa huomaamaan nopeasti epäselvistäkin osoitteista, mitä osoitteen sivulta voi löytyä. Botti voi myös kerätä tietoa siitä, että paljonko kukakin käyttäjä on puhunut kanavalla, tai vaikka kuinka usein jonkin sana on mainuttu.

4.2 Botin toteutus

Botin toteutuksen voi tehdä usealla tavalla. Tavanomaisin tapa on kuitenkin käyttää Eggdrop-ohjelmaa, joka on yksi helpommista tavoista toteuttaa botti [15]. Eggdropin kautta botin saa lähes valmiina käyttöön ja ominaisuuksia pystyy lisäämään botille käyttäen skriptejä, jotka ohjelmoidaan TCL-ohjelmointikieltä käyttäen [16]. Skriptejä löytyy myös valmiina muiden tekeminä. Tällaisen Eggdrop-botin pystyy ottamaan käyttöön kohtalaisen helposti ilman minkäänlaista ohjelmointitaitoa. Toinen yleinen tapa tehdä botti, on ohjelmoida se alusta asti omiin tarpeisiin sopivaksi. Tällöin ohjelmointikielenä voidaan käyttää esimerkiksi Pythonia [17] tai muita helposti laajennettavia skriptikieliä. Tämä kuitenkin vaatii paljon enemmän työtä ja ohjelmointitaitoa. TCL sekä Python ovat tulkattavia ohjelmointikieliä, jotka

tulkattavuutensa ansiosta sopivat hyvin bottien ohjelmointiin. Tulkattavuudella tarkoitetaan, että ohjelmaa voidaan muokata ja luoda dynaamisesti.

4.3 Eggdrop

Eggdrop on Robey Pointerin vuonna 1993 C-kielellä kirjoittama UNIX-järjestelmissä toimiva modulaarinen IRC-bottiohjelma [18]. Bottia pystyy kuitenkin komentamaan millä tahansa käyttöjärjestelmillä olevien asiakasohjelmien kautta. Eggdropin asetustiedoston saa ladattua internetistä, ja se on noin tuhannen rivin mittainen. Asetustiedosto pitää muokata omiin tarkoituksiin sopivaksi, eikä bottia pysty käyttämään ilman tiedoston muokkausta. Tiedoston muokkaukseen on kuitenkin hyvinkin yksityiskohtaisia ja helppoja ohjeita [15]. Muokkaus ei ole erityisen monimutkaista, ja kokenut bottien tekijä pystyy muokkaamaan asetustiedoston muutamassa minuutissa itselleen sopivaksi. Asetustiedosto sisältää myös käyttäjä- ja kanavatiedostot, joihin on tallennettu botille lisätyt käyttäjät ja kanavat. Kuviossa 8 pieni osa botin asetustiedostoa. Tässä kohtaa asetustiedostoa varmistetaan, että botin tekijä varmasti lukee asetustiedoston ja muokkaa sen haluamansa tyyppiseksi. Liitteessä 1 on esimerkki Eggdropin asetustiedoston vaatimista muutoksista.

```
# You MUST remove this line for your bot to start. This has been added to
# prevent you from starting up a bot that is not fully configured. Bots
# that have not been fully configured may join the wrong IRC network, the
# wrong channels, or generally do things that you do not want. Please make
# sure that you have double-checked every setting. There's also a similar line
# lower down, just to make sure you're reading :)
#die "Please make sure you edit your config file completely."
```

Kuvio 8. Asetustiedostoa läpikäydessä täytyy lukea tarkasti kohdat läpi. Lukemisen tarkkuutta tarkastellaan siten että välillä pitää kommentoida pois rivejä jotka aktiivisena ollessa sammuttavat botin.

4.4 Eggdropin ominaisuudet

4.4.1 Perusominaisuudet

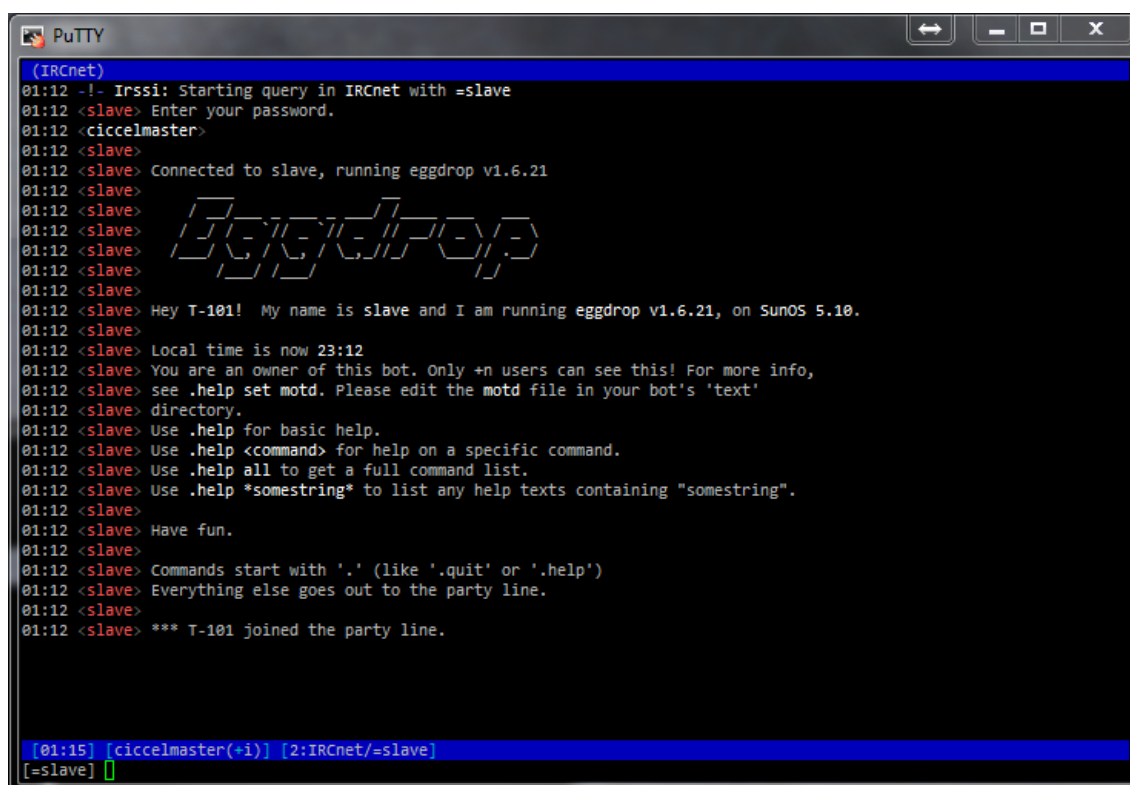
Eggdropissa on paljon valmiita perusominaisuuksia, jonka takia Eggdrop-botit ovatkin suosittuja. Valmiiden bottiohjelmien käyttämisessä etuna on oman vaivannäön

minimointi. Lähes kaikilla boteilla on kuitenkin tietyt perusominaisuudet, jotka toistuvat samoina. Eggdrop-bottien perusominaisuudet ovat Partyline, kanavanhallinta, jononhallinta sekä botnetit. Näin jokaisen bottia tekevän ei tarvitse laatia kaikkia botin ominaisuuksia alusta asti uudelleen, kun perustoiminnallisuuden saa Eggdropin kautta valmiina.

4.4.2 Partyline

Eggdropista valmiina ominaisuutena löytyy Partyline, jolla hallinnoidaan bottia. Partyline on IRC-verkosta irrallaan oleva yksityiskeskustelu käyttäjän ja botin välillä. Partyline toimii DCC-menetelmällä (direct client to client), joka on tuettu useimmissa käytetyissä IRC-asiakasohjelmissa. DCC-menetelmällä tarkoitetaan palvelua, joka käyttää suoria TCP-yhteyksiä siirtääkseen tietoa [10]. DCC-yhteydessä viestiliikenne ei toisin sanoen kulje IRC-verkon läpi, vaan asiakasohjelmat muodostavat keskenään toistensa välille yhteyden. Tässä ongelmaksi tosin voivat muodostua palomuurit tai muut verkon rakenteelliset seikat, jotka voivat estää yhteyden muodostamisia.

Botin tekijällä on oikeudet Partylineen, ja hän voi antaa siihen oikeuksia käyttäjille, jotka voivat kirjautua siihen salasanan avulla. Partylinessä voi keskustella muiden botin haltijoiden kanssa, sekä hallinoida kanavia joilla botti on käytössä. Bottiin voi asettaa public-komentoja, joita voidaan käyttää kanavakohtaisesti myös sellaisten käyttäjien toimesta, joilla ei ole oikeuksia Partylineen. Esimerkiksi operaattorin oikeuksien pyytäminen botilta on usein Partylinessä tehtävä komento, mutta säätietojen kysely botilta on kaikille sallittua. Jos botti antaisi operaattorin oikeuksia automaattisesti, riski siitä, että väärä henkilö saa oikeudet, kasvaa, kun salasanakyselyä ei vaadita. Säätietojen kysely taas puolestaan ei aiheuta tietoturvariskiä edes silloin kun kaikilla on siihen oikeus. Kuviossa 8 on slave-nimisen botin ja T-101-nimisen käyttäjän välinen Partyline yhteys. Käyttäjä T-101 on juuri ottanut yhteyden Partylineen. Partylinessä on ohjeita, kuinka bottia pystytään hallitsemaan katsomalla esimerkiksi help-tiedostoa.



```

(PuTTY)
01:12 -!- Irssi: Starting query in IRCnet with =slave
01:12 <slave> Enter your password.
01:12 <ciccelmaster>
01:12 <slave>
01:12 <slave> Connected to slave, running eggdrop v1.6.21
01:12 <slave>
01:12 <slave>
01:12 <slave>
01:12 <slave>
01:12 <slave>
01:12 <slave>
01:12 <slave> Hey T-101! My name is slave and I am running eggdrop v1.6.21, on SunOS 5.10.
01:12 <slave>
01:12 <slave> Local time is now 23:12
01:12 <slave> You are an owner of this bot. Only +n users can see this! For more info,
01:12 <slave> see .help set motd. Please edit the motd file in your bot's 'text'
01:12 <slave> directory.
01:12 <slave> Use .help for basic help.
01:12 <slave> Use .help <command> for help on a specific command.
01:12 <slave> Use .help all to get a full command list.
01:12 <slave> Use .help *somestring* to list any help texts containing "somestring".
01:12 <slave>
01:12 <slave> Have fun.
01:12 <slave>
01:12 <slave> Commands start with '.' (like '.quit' or '.help')
01:12 <slave> Everything else goes out to the party line.
01:12 <slave>
01:12 <slave> *** T-101 joined the party line.

[01:15] [ciccelmaster(+i)] [2:IRCnet/=slave]
[=slave]

```

Kuvio 9. Käyttäjä on avannut Partylinen slave-nimimerkkisen botin kanssa.

4.4.3 Kanavienhallinta

Kanavienhallinta on Eggdropin toinen tärkeä ominaisuus. IRC-kanavia voi hallita rajallisesti verkkojen ominaisuuksien avulla, kuten rajoittaa kävijämääriä ja kanavan näkyvyyttä. Nämä eivät kuitenkaan välttämättä aina riitä kanavien käytettävyyden ylläpitämiseen. Yksi tärkeimmistä kanavanhallinnallisista ominaisuuksista on rivitulvan (flood) estäminen. Rivitulvalla tarkoitetaan tilannetta, jossa yksittäinen henkilö tuottaa paljon tekstiä lyhyessä ajassa kanavalle tai vaihtoehtoisesti esimerkiksi vaihtelee nimimerkkiä tai poistuu ja saapuu kanavalle toistuvasti lyhyessä ajassa. Kuviossa 10 käyttäjä toistaa samaa komentoa toistuvasti ja näin aiheuttaa rivitulvalla poistamisen kanavalta. Nämä tilanteet koetaan häiritseväksi, koska ne hankaloittavat muuta keskustelua. Tämän takia boteille on usein määritelty rajoja määrästä, jonka verran käyttäjältä saa tulla tekstiä määrättyssä ajassa.

```

22:48 <@Tazze> !fact
22:48 <@Jonna^^> fact 522 of 605: kalmarin silmän halkaisia voi olla jopa 25cm.

22:48 <@Tazze> !fact
22:48 <@Jonna^^> fact 129 of 605: keskipainoisen miehen painoon tarvittaisiin
37 000 kimalaiskolibria.

22:48 <@Tazze> !fact
22:48 <@Jonna^^> fact 520 of 605: sarjakuvahahmo Karvisen syntymäpäivä on 19.6.

22:48 -!- Tazze was kicked from #Huulirasvariippuvuus by KiNGi [idiot! do not
repeat yourself!]

```

Kuvio 10. Itsensä toistaminen tai liian nopea tekstintuottaminen voivat aiheuttaa sen, että botti poistaa käyttäjän kanavalta. Yllä kuvattu käyttäytyminen voidaan kokea häiritseväksi. Kuvassa on myös esimerkki botin hauskoista käyttötavoista. Botin voi opettaa kertomaan erilaisia nippelitietoja.

Botin kanavan hallintaan kuuluu myös käyttäjän poistaminen kanavalta, jos esimerkiksi tällaista rivitulvaa tapahtuu. Asetustiedostossa voi myös määritellä muita ei-toivottuja käyttäytymiä, jotka aiheuttavat käyttäjän poistamisen. Esimerkiksi oikean nimen puuttuminen käyttäjän tiedoista saattaa olla osalla kanavista syy käyttäjän kanavalta poistamiseen. Botti voi myös automaattisesti isäntänimen ja identin avulla tunnistaa käyttäjän ja antaa tälle operaattorin oikeudet. Myös pelkän nimimerkinkin perusteella botin voi asettaa antamaan operaattorin oikeudet, mutta tässä tilanteessa on aina riski, että väärä henkilö tulee kanavalle tunnetulla nimimerkillä ja saa operaattorin oikeudet itselleen.

4.4.4 Jononhallinta

Eggdropissa on jononhallinta, joka auttaa esimerkiksi tilanteessa, jossa käyttäjän laatima skripti lähettää paljon tietoa IRC-verkkoon lyhyessä ajassa. Tilanteessa, jossa käyttäjä pyytää botilta monta kertaa jonkin tiedon, kuten säätilan usealta paikkakunnalta, botti käyttää myös jononhallintaa. Tässä tilanteessa jononhallinnan ansiosta botti ei syötä tietoja verkkoon reaaliajassa, vaan odottaa, että on kulunut sopiva aika ennen kuin syöttää seuraavan paikkakunnan tiedot käyttäjän nähtäväksi. Jononhallinnassa botti tarkastelee, onko sama tieto jo lähetetty aikaisemmin verkkoon ja onko kulunut riittävästi aikaa edellisestä viestistä, jottei tekstiä tulkita rivitulvaksi. IRC-palvelimet myös poistavat botin verkosta, mikäli rivitulvaa tulee liikaa ja tämän takia botti menettää yhteyden palvelimeen. Jos botilta kysytään nopeassa ajassa useita kysymyksiä, joihin se vastaisi reaaliaikaisesti ilman jononhallintaa, se tuottaisi verkkoon

niin paljon tekstiä, että IRC-verkko tulkitsisi sen häirinnäksi ja poistaisi botin palvelimelta.

4.4.5 Botnetit

Eggdropia käytetään myös Botnettien tekemiseen. Botnetit ovat IRC-bottien verkkoja, joissa useampi botti liitetään yhteen. Botnetejä käytetään esimerkiksi vikasietoisuuden toteuttamiseksi. Yhden botin menettäessä yhteyden verkkoon toinen botti voi ottaa hoitaakseen ensimmäisen botin tehtävät. Botnetissä olevat botit voivat myös kutsua toisiaan suljetuille kanaville ja jopa käynnistää uuden botin käyttöön, jos yhteys johonkin bottiin menetetään. Botnetin käyttämisessä etuna on, että botit voivat jakaa olemassa olevat tiedot keskenään, eikä tällöin muodostu konflikteja bottien välille, mikä on mahdollista käyttäessä botteja, joilla ei ole keskinäistä yhteyttä.

Botnetejä voidaan käyttää myös haitalliseen toimintaan. Esimerkkeinä voidaan mainita hajautettujen palvelunestohyökkäysten toteuttaminen, vakoilutiedon kerääminen ja roskapostien välittäminen. Botneteistä voi muodostaa niin sanottuja zombi-armeijoita, joita isäntäkoneet käskyttävät. Zombeilla on myös mahdollisuus välittää isännälle tietoa hallintakoneiden kautta. Tyypillisesti botnettien kontrollikanavana käytetään IRC:tä. Käyttäjä ei yleensä tiedä koneellansa olevan haittaohjelmaa, minkä seurauksena kone päätyy botnetin zombi-armeijaan.[19].

4.5 TCL ja skriptit

Pelkkä Eggdrop-botti ilman mitään lisäyksiä ei tarjoa käyttäjälle merkittävää lisäarvoa, vaikka asetukset olisivatkin kanavan mukaiset. Bottien hyödyllisyys tulee esille vasta, kun bottiin lisätään skriptejä, joilla saadaan botille käyttötarkoitukseen sopivia ominaisuuksia. Bottien ominaisuudet voivat olla niin viihteellisiä kuin myös hyödyllisiä. Eggdropille tehdyt skriptit ohjelmoidaan TCL-kielellä. TCL on John Ousterhoutin luoma tulkattava ohjelmointikieli [16]. Tulkattava ohjelmointikieli suorittaa ohjelmointikielisiä lauseita yksi kerrallaan toisin kuin käännettävät ohjelmointikielet, joissa koko ohjelma käännetään konekieliseksi ennen suoritusta. Lähdekoodin tulkkauksen on hitaampaa kuin kääntäminen. Tulkkattu kieli kuitenkin antaa mahdollisuuden muokata ja luoda koodia

dynaamisesti. TCL:ää käytetään Eggdrop-bottien skriptien luomisen lisäksi usein nopeiden prototyyppien luomiseen ja testaukseen.

Botit tarjoavat käyttäjille lisää sisältöä skripteillä tuotettujen ominaisuuksien avulla. Usein kanavien botit ovat pääasiassa tuottamassa iloa kanavien käyttäjille, mutta pitävät samalla kanavien järjestystä yllä. Viihteellisenä toimintona boteilla on usein esimerkiksi listoja, joista näkee kuka on puhunut kanavalla eniten tai vaikkapa kanavalla sanottuja hauskoja lauseita, jotka ovat usein kontekstista irrallaan. Hyödyllisiä boteilla käytettyjä ominaisuuksia edustaa esimerkiksi sääpalvelu, josta näkee pienellä viiveellä säätilan paikkakuntaakohtaisesti. Toinen usein boteilla oleva palvelu on ilmoitus vuorokauden vaihtuessa päivämäärästä ja viikosta sekä kyseisen päivän nimipäiväsankareista. Tämän tiedon voi myös saada myöhemmin botilta kysymällä, kuten kuviossa 11 on tehty.

```
22:45 <@Tazze> !pvm
22:45 <@KiNGi> Tänään on Tiistai 20.3.2012 (Viikko 12) ja nimipäivää viettää
Aki, Joakim, Kim, Jaakkima.
22:45 <@Tazze> !sää helsinki
22:45 <@KiNGi> ie 3 Helsinki, Pirkkola, mitattu klo 22:23: 2.0°C, poutaa. Tien
lämpötila 0.6°C, tien pinta kuiva.
```

Kuvio 11. Botilta voi kysyä esimerkiksi päivämäärä- tai sää tietoja.

5 Skriptien käyttö

5.1 Bottien hyödyntäminen

IRC:ssä käytetään useilla kanavilla botteja apuna pitämään kanavan järjestystä yllä. Lisäksi boteilla voi olla myös erilaisia ominaisuuksia joita voi käyttää apuna tiedon keräämiseen ja tallentamiseen. Tiedon keruuseen botit ovat hyvin tarpeellisia, koska IRC:ssä käsiteltävät asiat unohtuvat helposti, kuten tapahtuu myös kasvotusten käydyissä keskusteluissa. Tietojen kerääminen ja tallentaminen botille helpottaa myös niitä, jotka eivät juuri keskustelun aikaan ole paikalla vaan haluavat jälkikäteen lukea keskustelun pääasiat.

Botille voi luoda loputtomasti erilaisia ominaisuuksia. Vain mielikuvitus ja ohjelmointitaidot ovat rajoituksena sille, mitä botti voi tehdä. Valmiita skriptejä löytyy internetistä paljon, mutta niitä joutuu usein hieman muokkaamaan omiin tarpeisiinsa, sekä mahdollisesti muutenkin tarkistamaan tietoturva-aukkojen välttämiseksi [20].

5.2 "Mitä tarvitsee tehdä"- skriptin esittely

Botti on myös hyvä työväline muistutuksille itselleen tai käyttäjille laajemmalti. Voidaan esimerkiksi tehdä "mitä tarvitsee tehdä" -lista. Lista toimii vaikka siten, että kirjoittamalla "!todo Maarit muista tehdä päättötyö", tallentaa botti tämän ja kysymällä botilta "!todo_list Maarit", kertoo botti listalla olevat asiat. Listasta voi poistaa kohtia komennolla "!todo_poista Maarit 1". Komento poistaa listalla ensimmäisenä olevan. Laittamalla komenttoon numeron 1 paikalle 2 poistuu listalla toisena oleva tehtävä. Lisäksi kirjoittamalla "!todo listat" se näyttää listan kaikista nimimerkeistä, joita botille on luotu. Botilta saa myös poistettu nimimerkin komennolla "!todo_tuhoa Maarit". Kuviossa 12 esitellään kyseisen skriptin toimintaan SlaveBotilla. Liitteessä 2 on esitetty "mitä tarvitsee tehdä"-listan toteuttamiseksi tarvitta skripti.

```
21:22 -!- SlaveBot [~slavebot@siilo.netcrew.fi] has joined #inssiharjoitus
21:22 < Tassukka> !todo_help
21:23 < Tassukka> !todo_list Tassukka
21:25 < Tassukka> !todo Tassukka kirjoita todo-lista
21:25 < Tassukka> !todo Tassukka kastele kukat
21:26 < Tassukka> !todo Tassukka muista ostaa kissanruokaa
21:27 < Tassukka> !todo Tassukka osta hammastahnaa
21:27 < Tassukka> !todo_list Tassukka
21:28 < Tassukka> !todo_list Tassukka
21:28 < SlaveBot> tassukka[1]: oeoeeoeoe
21:28 < SlaveBot> tassukka[2]: kirjoita todo-lista
21:28 < SlaveBot> tassukka[3]: kastele kukat
21:28 < SlaveBot> tassukka[4]: muista ostaa kissanruokaa
21:28 < SlaveBot> tassukka[5]: osta hammastahnaa
21:28 < Tassukka> !todo_poista Tassukka 1
21:28 < Tassukka> !todo_list Tassukka
21:28 < SlaveBot> tassukka[1]: kirjoita todo-lista
21:28 < SlaveBot> tassukka[2]: kastele kukat
21:28 < SlaveBot> tassukka[3]: muista ostaa kissanruokaa
21:28 < SlaveBot> tassukka[4]: osta hammastahnaa
```

Kuvio 12. Botin nimenä kyseisessä esimerkissä SlaveBot ja Tassukka nimimerkillä oleva käyttäjä tekee botille listan tehtävistä asioista.

Liitteessä 2 esitellään "Mitä tarvitsee tehdä" -skripti, jonka ohjelmaa tarkastellaan tässä lähemmin. Skriptin alussa määritetään (engl. set) käytettävät tiedostot "Mitä tarvitsee tehdä"-skriptin tallennuksia varten.

```
set learn(file) "learn.dat"
set learn(tempfile) "learn.tmp"
```

Lisäksi määritetään ne kanavan rivit, joihin botin halutaan reagoivan jollakin tavalla (engl. triggers). Muut rivit botti ohittaa ilman, että se reagoi niihin.

```
set learn(forget) "!todo_tuhoa"
set learn(insert) "!todo"
set learn(del) "!todo_poista"
set learn(view) "!todo_lista"
set learn(wordlist) "!todo_listat"
set learn(help) "!todo_help"
```

Tämän jälkeen voidaan vaihtaa botin käyttäymistä tulosteeseen sen osalta, tulevatko pyynnöt kanavalle vai käyttäjälle yksityisenä viestinä (engl. method). Kun käyttäjä kysyy botilta tietoa, tässä kohtaa voidaan määrittää, tulevatko tiedot käyttäjälle yksityisenä viestinä vai kaikille näkyviin kanavalle.

```
set learn(method) "NOTICE"
set learn(method-def) "PRIVMSG"
```

Tähän asti olevat kohdat voi käyttäjä määritellä haluamallaan tavalla. Käyttäjä voi siis näitä arvoja vaihtamalla vaikuttaa reagoitaviin lauseisiin sekä siihen, mihin botin vastaukset tulevat. "Mitä tarvitsee tehdä"-listan tallennussijainnin voi asettaa olemaan yrityksen varmuuskopioidulla verkkolevyllä, jolloin listan häviämisestä ei ole pelkoa.

Seuraavien kohtien muuttamiseen ei ole tarvetta, koska ne eivät vaikuta käytettävyyteen. Tämän jälkeen ohjelmassa sidotaan (engl. bind) määritetyt reagoitavat rivit kuunneltaviksi ja yhdistetään ne funktioon, joka reagoi löydettyyn lauseeseen.

```
bind pub - $learn(forget) learn_forget
bind pub - $learn(insert) learn_insert
bind pub - $learn(del) learn_del
bind pub - $learn(view) learn_view
bind pub - $learn(wordlist) learn_list
bind pub - $learn(help) learn_help
```

Skripti myös latautuessaan kirjoittaa lokiin tiedon onnistuneesta käynnistymisestä.

```
putlog "Loaded TODO"
```


Tämän jälkeen varmistetaan, että tallennukseen käytettävään tiedostoon on oikeus kirjoittaa.

```
if {[file exists $learn(file)]} {set fxtmp [open $learn(file)
w] ; close $fxtmp}
```

Näiden jälkeen on hakusanan vertailun tekevä apufunktio ja lopuksi reagoitaviin riveihin sidotut funktiot.

Seuraavaksi esitellään tarkemmin muutamaa skriptin toiminnallisuuden osaa. Keskeisimmät toiminnot ovat tehtävän lisääminen, poistaminen sekä listaaminen.

Skriptin alussa on:

```
set learn(insert) "!todo"
set learn(del) "!todo_poista"
set learn(view) "!todo_lista"
```

Nämä määrittelevät komennot, joilla käyttäjä komentaa bottia. Botti hyödyntää näitä sitten seuraavissa riveissä:

```
bind pub - $learn(insert) learn_insert
bind pub - $learn(del) learn_del
bind pub - $learn(view) learn_view
```

Bind-komennolla sidotaan muuttujan (esimerkiksi \$learn(del)) sisältö (tässä tapauksessa "!todo_poista") botin kuunneltavaksi sekä asetetaan funktio, jota kutsutaan täsmäävän komennon. Poiston (engl. delete, lyhennettynä del) tapauksessa funktio on siis learn_del.

Huomataan learn_insert –funktion olevan määritelty seuraavasti:

```
proc learn_insert {nick host hand chan text}
```

Eggdrop antaa automaattisesti funktiolle argumentit, jotka on lueteltu aaltosulkeissa. Käytännössä siis rivin kirjoittaneen käyttäjän nimimerkin, isäntänimen, identin, kanavan, sekä itse kirjoitetun rivin. Funktiossa tarkistetaan aluksi, että tarvittavat tiedot, eli tehtävän kohde, sekä tehtävä itse, on annettu. Mikäli näin ei ole, ilmoitetaan syntaksivirheestä ja lopetetaan funktion suorittaminen. Muussa tapauksessa jatketaan suorittamista eli lisätään tehtävä väliaikaistiedoston avulla tallennettuun tehtävälistaan ja ilmoitetaan lisäyksen onnistuneen. Ilmoituksen tapa on skriptin alussa määritelty (engl. method).

Funktio `learn_del` noudattaa argumenttien ja funktiomäärittelyn osalta samaa kaavaa kuin `learn_insert`. Suorituksen osalta se myös tarkistaa, että käskytyksessä on annettu tehtävän kohteen nimimerkki, sekä tehtävän numero. Mikäli tehtävää ei löydy, ilmoitetaan siitä ja lopetetaan funktion suoritus. Muutoin poistetaan tehtävä jälleen väliaikaistiedoston avulla ja ilmoitetaan määritellyllä tavalla poiston onnistumisesta.

Myös `learn_view` noudattaa hyvin samankaltaista kulkua. Tarkistetaan, että on annettu tehtävän kohde, ja jollei sitä ole, ilmoitetaan ja lopetetaan suoritus. Mikäli yhtään tehtävää ei kohteella ole, ilmoitetaan ja lopetetaan suoritus. Muutoin tulostetaan kaikki täsmäävät tehtävät.

5.3 Muita esimerkkejä botille tehtävistä skripteistä

5.3.1 Botin toiminta kanavan vartijana

Usein tärkeimpänä ominaisuutena botille on ylläpitää kanavan järjestystä siten, että tietyillä vastuullisilla henkilöillä pysyy operaattorin oikeudet. Tämä toiminto on usein tehty botille niin, että operaattorin oikeudet voi pyytää botilta yksityisellä viestillä. Tällöin tunnistautumiseen tarvitaan salasana. Vaihtoehtoisesti operaattorin oikeudet pyydetään suoraan kanavalta, ja botti tunnistaa henkilön identin ja isäntänimen perusteella. Usein käytetty komento kyseiselle toiminnolle on `!op`. Botti voi myös jakaa operaattorin oikeuksia halutuille käyttäjille heti heidän saavuttuaan kanavalle, tällöin erillistä pyyntöä oikeuksien saamiseksi ei tarvita. Näissäkin tapauksissa botti tunnistaa käyttäjän identin ja isäntänimen perusteella.

Operaattorin oikeudet ovat tärkeitä, sillä operaattorin oikeudet omaava henkilö voi poistaa häiritseviä käyttäjiä kanavalta ja pystyy näin hallitsemaan kanavaa. Operaattorin oikeuksilla pystytään myös jakamaan kanavalla puheoikeuksia. IRC:ssä on ominaisuutena valvottu keskustelu (`moderate`), jolloin ainoastaan operaattorin oikeudet tai puheoikeuden (`voicen`) omaavat käyttäjät voivat puhua kanavalla. Tämä on käytännöllistä esimerkiksi, jos IRC:ssä halutaan pitää kokous. Tällöin keskustelu saadaan varmemmin pidettyä aiheessa, kun vain yhdellä käyttäjällä kerrallaan on puheoikeus kanavalla.

5.3.2 Botti apuna ajanhallinnassa

Botin voi asettaa muistuttamaan tiettyä kellonaikana haluttu asia. Botille kerrotaan kellonaika, milloin botin pitää reagoida, jolloin botti esimerkiksi laittaa viestin joko yksityiseen ikkunaan tai kanavalle. Esimerkiksi jos sovitaan asiakkaan kanssa, että hänelle soitetaan 14:00, voi botin kätkeä ilmoittamaan 13:55 "Muista soittaa asiakkaalle Nimi, numero, ticketinnumero". Tästä löytyy esimerkki kuviossa 13.

```
22:39 <@Tazze> !alarm add 22:40 Tazze Muista Soittaa asiakkaalle!
22:39 <@Jonna^> Alarm "Tazze Muista Soittaa asiakkaalle!" (22:40) added to
list. Time to alarm 1 minutes.
22:40 <@Jonna^> Alarm! Tazze Muista Soittaa asiakkaalle!
```

Kuvio 13. Botin voi asettaa muistuttamaan asiasta, jolloin ei tarvitse itse aktiivisesti muistaa kaikkea.

Botille voi myös tehdä listoja tulevista tapahtumista. Esimerkiksi tulevia koulutuksia tai alan tapahtumia voi listata bottiin ja määrättyllä komennolla listan saan näkyviin käyttäjän niin halutessa. Botin voi ohjelmoida siten, että se automaattisesti tarkastaa menneet tapahtumat pois listalta.

Bottia voi käyttää myös kellokorttina töissä. Botiin voi esimerkiksi kirjoittaa "!töissä", ja botti kirjaa kirjauksen kellonajan järjestelmään. Töiden päättyessä ilmoitetaan vastaavasti "!poistu" ja botti kirjaa taas kellonajan muistiin. Jos käytössä on esimerkiksi liukuva työaika, pystyy itse helposti tarkistamaan myöhemmin, moneltako on tullut töihin. Botin voi myös opettaa laskemaan liukumia. Botin huomatessa, että sisäänkirjautumisen ja uloskirjautumisen välinen aika on sovittua työaikaa pidempi tai lyhyempi, ilmoittaa botti työntekijälle, paljonko liukuman saldo on. Botti voidaan asettaa esimerkiksi huomauttamaan, mikäli plus- tai miinustunteja on kertynyt liukumaan liikaa, ja halutessa ilmoittamaan asiasta esimiehelle.

5.3.3 Osoitekirja botille

Botin avulla voi toteuttaa osoitekirjan, jonka avulla käyttäjä voi helposti tarkastaa nimimerkin yhteystiedot. Botilta voi kysyä esimerkiksi "!kuka MaijjaMOB" ja botti antaa vastaukseksi "Maija Meikäläinen Mobiiliosaaaja maija.meikalainen@osoite.fi 050 123 4567". Hakemisen voi tehdä myös siten, että botille kerrotaan oikea nimi, ja se kertoo

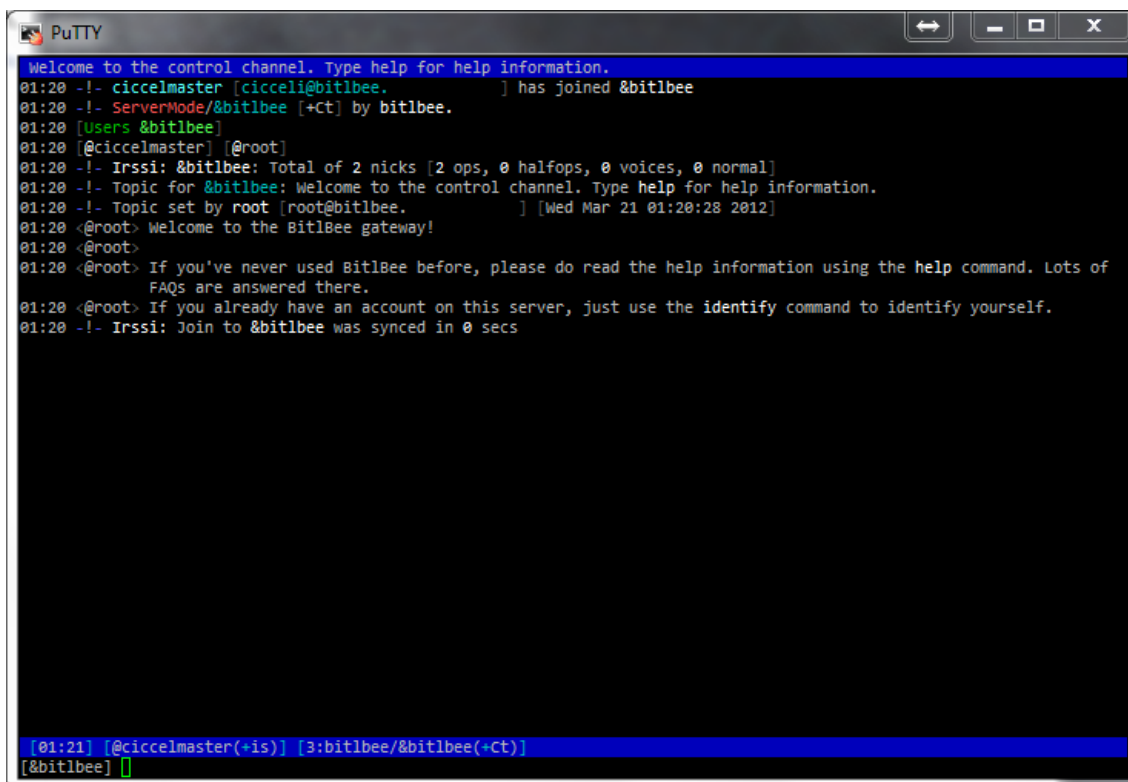
vastineeksi nimimerkin. Sopivassa järjestelmässä botin voi asettaa hakemaan tiedot yrityksen omasta osoitekirjasta.

5.3.4 Tiketöintijärjestelmän yhdistäminen bottiin

Jos botin yhdistää tikettijärjestelmään, voi botti ilmoittaa, kun tikettijärjestelmään tulee uusi ticketti, ja botti voi samalla osoittaa ticketin jollekin osaajalle. Esimerkiksi uuden mobiiliticketin saapuessa järjestelmään botti ilmoittaisi ”tikettinumero saapui järjestelmään MaijaMOB huomio”, jolloin MaijaMOBilla näkyy korostettuna kyseinen ilmoitus, kun hänen nimimerkinsä mainitaan. Tikettijärjestelmään yhdistettynä olevalta botilta voi myös esimerkiksi kysellä, kuinka monta tickettiä jonossa on, tai jos tiketeissä on aikavaatimus, voi botin ohjelmoida ilmoittamaan, kun ticketin aikaraja lähestyy.

5.4 IRC:n laajentaminen muihin palveluihin

Erilaisilla lisäpalveluilla on mahdollista yhdistää IRC:hen muitakin palveluita. Esimerkiksi BitlBeen [21] avulla Facebookin keskustelun saa yhdistettyä IRC-asiakasohjelmalle. BitlBee on ohjelma, jonka avulla voidaan liittää erilaisia pikaviestimien viestejä toisesta palvelusta toiseen. Asentamalla BitlBeen palvelimelle saa IRC:hen hallintaikkunan, jossa näkyy kaikki Facebookissa paikalla olevat käyttäjät ja heille pystyy viestittämään IRC:stä Facebookin keskusteluun. Bitlbeen ohjauspaneelista löytyy kuva kuviossa 14. BitlBee tukee myös seuraavia palveluita: XMPP/Jabber, MSN Messenger, Yahoo! Messenger, AIM, ICQ sekä Twitter. Vastaavalla palvelulla voisi saada myös yrityksissä usein käytetyn Microsoft Office Communicatorin yhdistettyä IRC:hen, minkä seurauksena vaihto MOC:n käytöstä IRC:n käyttöön helpottuisi, kun voi käyttää molempia pikaviestimiä ristiin vaihdon ajan.



```

Welcome to the control channel. Type help for help information.
01:20 -!- ciccelmaster [cicceli@bitlbee.      ] has joined #bitlbee
01:20 -!- ServerMode/#bitlbee [+Ct] by bitlbee.
01:20 [Users #bitlbee]
01:20 [ciccelmaster] [root]
01:20 -!- Irssi: #bitlbee: Total of 2 nicks [2 ops, 0 halfops, 0 voices, 0 normal]
01:20 -!- Topic for #bitlbee: Welcome to the control channel. Type help for help information.
01:20 -!- Topic set by root [root@bitlbee.      ] [Wed Mar 21 01:20:28 2012]
01:20 <@root> Welcome to the BitlBee gateway!
01:20 <@root>
01:20 <@root> If you've never used BitlBee before, please do read the help information using the help command. Lots of
FAQs are answered there.
01:20 <@root> If you already have an account on this server, just use the identify command to identify yourself.
01:20 -!- Irssi: Join to #bitlbee was synced in 0 secs

[01:21] [ciccelmaster(+is)] [3:bitlbee/#bitlbee(+Ct)]
[bitlbee]

```

Kuvio 14. BitlBeen ohjauspaneelistä kuva. Tätä kautta saa yhdistettyä sosiaalisia medioita ja muita pikaviestimiä IRC:hen.

Bottien avulla IRC:n saa myös liitettyä esimerkiksi MediaWikiin [32, s.4]. MediaWiki on ohjelmisto yhteisöllisten materiaalin julkaisuun. Tätä voi käyttää esimerkiksi siten, että botille laittetaan asioita, kuten lomat tai kokoukset, ja se siirtää ne MediaWikiin. Tällöin vaikka botista poistuisikin tieto loman jälkeen asiasta, jää tieto MediaWikin puolelle ja näin tietoa voi tarkastella myös jälkeenpäin.

Botin avulla viestejä voi lähettää myös sähköpostiin tai matkapuhelimeen. Viestejä voi lähettää sekä matkapuhelimesta botille että botin kautta muihin matkapuhelimiin. Tämä vaatii botin olevan yhteydessä johonkin matkapuhelimeen, joka välittää viestit. Puhelin on liitetty palvelimeen sarjaportilla tai USB-liittimellä. Operaattorit tarjoavat myös palvelua, jonka avulla tekstiviestien vastaanotto ja lähettäminen on mahdollista suorittaa ilman tarkoitukseen varattua matkapuhelinta. [33, s. 20–22].

Yrityksen verkkosivulle voi myös asettaa suoran linkin selainpohjaiseen IRC asiakasohjelmaan. Tällöin asiakkaiden on helppo käyttää yrityksen IRC-palveluita ilman suurta vaivaa opetella käyttämään IRC:tä. Painaessaan yrityksen sivulla olevasta

linkistä asiakas siirtyy suoraan yrityksen ohjaamalle kanavalle. Kanavan botin voi asettaa antamaan ohjeita kaikille yrityksen sivun linkin kautta tuleville.

6 Yhteenveto

Työssä on käsitelty IRC:n käyttöä yritysten viestinnässä ja pohdittu sen hyviä sekä huonoja puolia. Työssä päädyttiin siihen, että IRC on parhaimmillaan sellaisissa tilanteissa, joissa useat henkilöt haluavat puhua samasta asiasta. IRC:n hyödyt tulevat esille myös silloin, kun tietoa pitää jakaa myös useille henkilöille, jotka eivät juuri sillä hetkellä ole koneen ääressä. Tilanteissa, joissa ei välttämättä ole varma, kuka osaa vastata kysymykseen, IRC:n edut tulevat esille, kun kysymyksen pystyy esittämään samalla kertaa kaikille. Hankalimmaksi asiaksi IRC:n käyttöönotossa todettiin vaikeus oppia käyttämään IRC:n kaikkia mahdollisuuksia. Lisäksi vaikeutta käyttöönottoon voi tuoda käyttäjien haluttomuus oppia uutta.

Tietoturva on erittäin oleellinen osa yritysten toiminnassa ja siihen pitää erityisesti kiinnittää huomiota kaikessa viestinnässä. Työssä on pohdittu IRC:n tietoturvaongelmia ja ratkottu niitä esimerkiksi siirtämällä yrityksen IRC-keskustelut IRC-verkkoon, johon vain yrityksen työntekijät pääsevät. Lisäksi inhimillisten virheiden mahdollisuutta voi vähentää insinööriyössä esitetyin keinoin, kuten pitämällä julkisen IRC-verkon kanavat erillisessä asiakasohjelmassa kuin sisäisen verkon kanavat. Tärkeintä on kuitenkin huolellisuus ja tietämys yrityksen tietoturvapoliitika.

IRC-bottien hyödyntäminen yrityksen viestinnässä todettiin kannattavaksi etenkin, kun botille luodaan mahdollisimman hyvät ja yrityksen toimintaa tukevat skriptit. Työssä esiteltiin skriptien tekemistä yhden skriptin avulla sekä kerrottiin erilaisia vaihtoehtoja, mitä skriptien avulla voidaan tehdä. Esimerkkejä oli vain muutama mutta tarpeiden ja halujen mukaan skriptejä löytyy sekä valmiina internetistä että niitä voi tehdä alusta asti itse. Skriptien kohdalla ensin on tärkeintä selvittää, mitä botin halutaan tekevän ja sen jälkeen lähteä selvittämään, miten botille halutut ominaisuudet saadaan.

”Mitä tarvitsee tehdä”-skriptin avulla insinööriyössä esitellään skriptien laatimista ja käyttöä. Liitteestä 2 löytyvä skriptin toimintalogiikkaa on selvitetty tarkentavasti. Kyseinen skripti on tehty sanakirja-skriptin pohjalta. Tämä osoittaa hyvin, ettei kaikkea

tarvitse tehdä alusta asti tyhjästä. Skriptejä löytyy internetistä paljon ja hieman soveltamalla niistä saa omien tarpeidensa mukaisia.

Automatisoinnilla voidaan helpottaa yrityksissä viestintää, siten ettei kaiken tarvitse tapahtua käyttäjien toimesta. Bottien avulla pystyy hoitamaan osan tehtävistä. Virheiden määrä vähenee, kun inhimillisen virheen mahdollisuuksia vähennetään. Lisäksi esimerkiksi asiakaspalvelun puolella voidaan helpottaa työtaakkaa siirtämällä osa tehtävistä bottien vastuulle.

Lähteet

- 1 Legal Procedure against "Facebook Ireland Limited". 2011. Verkkodokumentti. Europe versus Facebook. <<http://europe-v-facebook.org/EN/Complaints/complaints.html>>. Luettu 21.3.2012.
- 2 RFC 1459: Internet Relay Chat Protocol. 1993. Verkkodokumentti. J. Oikarinen, D. Reed. <<http://www.rfc-editor.org/rfc/rfc1459.txt>>. Luettu 21.3.2012.
- 3 Marianne Pyra; Using Internet Relay Chat. Que Corporation 1995, ISBN 0-7897-0020-4
- 4 IRC Networks - Summary. 1998. Verkkodokumentti. netsplit.de-palvelu. <<http://irc.netsplit.de/networks/summary.php>>. Luettu 20.3.2012.
- 5 IRC-ohje: Irssi ja jalava. 2010. Verkkodokumentti. Jyväskylän fyysikkokerho. <<http://groups.jyu.fi/fk/irssi>>. Luettu 15.3.2012.
- 6 mIRC: Internet Relay Chat client. 1995. Verkkodokumentti. mIRC Co. Ltd. <www.mirc.com>. Luettu 1.3.2012.
- 7 HydraIRC: The Professional IRC Client. 2003. Verkkodokumentti. The HydraIRC Community. <www.hydrairc.com>. Luettu 1.3.2012.
- 8 Irssi - The client of the future. 2000. Verkkodokumentti. The Irssi Project. <<http://irssi.org>>. Luettu 1.3.2012.
- 9 PuTTY: A Free Telnet/SSH Client. 2011. Verkkodokumentti. Simon Tatham. <<http://www.chiark.greenend.org.uk/~sgtatham/putty>>. Luettu 1.3.2012.
- 10 New IRCNet Changes. 1999. Verkkodokumentti. Thomas Kuiper. <<http://www.irchelp.org/irchelp/ircd/ircnet210.html>>. Luettu 15.3.2012.
- 11 Turban Efraim, Leidner Dorothy, McLean Ephraim. Wetherbe James; Information Technology for Management: Transforming Organizations in the Digital Economy 5. painos. John Wiley & Sons 2005, ISBN 0-471-70522-5
- 12 Haastattelu, Antti Kiuru, Viestintävirasto, Maaliskuu 2012.
- 13 RFC 6101: The Secure Sockets Layer (SSL) Protocol Version 3.0. 2011. Verkkodokumentti. A. Freier, P. Karlton, Netscape Communications, P. Kocher, Independent Consultant. <<http://www.rfc-editor.org/rfc/rfc6101.txt>>. Luettu 21.3.2012.
- 14 EFnet IRC Network Forum. 2007. Verkkodokumentti. EFnet Network. <<http://forum.efnet.org/ssl-test-t2131.html>> Luettu 20.3.2012.
- 15 Slennox's egghelp page. 1999. Verkkodokumentti. egghelp.org-palvelu. <<http://www.egghelp.org>> Luettu 15.3.2012.

- 16 Paul Raines, Jeff Tranter; Tcl/Tk in a Nutshell. O'Reilly Media, Inc. 1999, ISBN: 978-1-56592-433-8
- 17 Mark Summerfield; Programming in Python 3: A Complete Introduction to the Python Language 2. painos. Addison-Wesley 2009, ISBN: 0-321-68056-1
- 18 Eggheads.org Our Producter. 1998. Verkkodokumentti. Egghead.org. <<http://www.eggheads.org/products/>> Luettu 15.3.2012.
- 19 Introduction | The HoneyNet Project. 2008. Verkkodokumentti. The HoneyNet Project. < <http://www.honeynet.org/node/51>> Luettu 20.3.2012.
- 20 TCL Scripts for Eggdrop IRC Bots. 2004. Verkkodokumentti. Cell. <<http://cell.cubicle.fi/scripts>>. Luettu 20.3.2012
- 21 BitlBee, An IRC to other chat networks gateway. 2012. Verkkodokumentti. Wilmer, Jelmer, Maurits. <<http://www.bitlbee.org>>. Luettu 15.3.2012.
- 22 Chris Brogan; Google+ for Business: How Google's Social Network Changes Everything. Que 2011, ISBN:978-0-7897-4914-7
- 23 E. A. Vander Veer; Facebook: The Missing Manual. O'Reilly Media Inc. 2011, ISBN: 978-1-4493-9741-8
- 24 Jamie Turner; How to Make Money Marketing Your Small Business on Twitter. FT Press 2010, ISBN: 978-0-13-261540-2
- 25 Joe Schurman; Microsoft® Voice and Unified Communications. Addison-Wesley Professional 2009, ISBN: 978-0-321-57995-9
- 26 Harry Max; Taylor Ray; Skype: The Definitive Guide. Que 2006, ISBN: 978-0-321-40940-9
- 27 Michael Miller; Special Edition Using the Internet and Web. Que 2001, ISBN:978-0-7897-2613-1
- 28 Tietotekniikan termitalkoot. 2007. Verkkodokumentti. Sanastokeskus TSK ry. <<http://www.tsk.fi/tsk/termitalkoot/>> Luettu 4.4.2012.
- 29 NickServ allows you to. Verkkodokumentti. 2006. TechNerd. <<http://www.technerd.net/nickserv.html>> Luettu 18.4.2012.
- 30 XS4ALL | Webchat. 2008. Verkkodokumentti. XS4ALL.<<http://webchat.xs4all.nl/>> Luettu 18.4.2012.
- 31 Manish Gupta; Raj Sharman; Social and Human Elements of Information Security: Emerging Trends and Countermeasures. IGI Global 2008, ISBN: 978-1-60566-036-3
- 32 Daniel J. Barrett; MediaWiki. O'Reilly Media, Inc. 2008, ISBN: 978-0-596-51979-7
- 33 Nicola Pero; SMS Messaging Applications, O'Reilly Media, Inc. 2009. ISBN: 978-0-596-55809-3

Eggdropin asetustiedosto

```
##### BASIC SETTINGS #####
# Osa asetustiedoston tiedoista on poistettu koska ne toimivat
# perusasetuksissa ilman muutoksia. Kohdat on merkitty seuraavasti
# Kohdassa olevat tiedot ovat toimivat perusasetuksilla
# This setting defines the username the bot uses on IRC. This setting has
# no effect if an ident daemon is running on your bot's machine.
set username "slavebot"
# Kohdassa olevat tiedot ovat toimivat perusasetuksilla
set admin "Maarit <email: maarit.klami@metropolia.fi>"
set network "I.dont.have.a.botnet"
set timezone "UTC+2"
set offset "-2"
##### LOG FILES #####
# Kohdassa olevat tiedot ovat toimivat perusasetuksilla
##### CONSOLE #####
# Kohdassa olevat tiedot ovat toimivat perusasetuksilla
##### FILES AND DIRECTORIES #####
# Specify here the filename your userfile should be saved as.
set userfile "slavebot.user"
# Specify here the filename Eggdrop will save its pid to. If no pidfile is
# specified, pid.(botnet-nick) will be used.
#set pidfile "pid.LamestBot"
# If you want your userfile to be sorted upon saving, enable this setting.
# This causes the bot to use a bit more CPU when saving the userfile.
set sort-users 1
# Kohdassa olevat tiedot ovat toimivat perusasetuksilla
##### BOTNET/DCC/TELNET #####
# Kohdassa olevat tiedot ovat toimivat perusasetuksilla
##### MORE ADVANCED SETTINGS #####
# Kohdassa olevat tiedot ovat toimivat perusasetuksilla
##### MODULES #####
```

```
# Kohdassa olevat tiedot ovat toimivat perusasetuksilla
set mod-path "modules/"
loadmodule blowfish
loadmodule dns
#### CHANNELS MODULE ####
# Kohdassa olevat tiedot ovat toimivat perusasetuksilla
loadmodule channels
# Enter here the filename where dynamic channel settings are stored.
set chanfile "slavebot.chan"
# Kohdassa olevat tiedot ovat toimivat perusasetuksilla
#### SERVER MODULE ####
# This module provides the core server support. You have to load this
# if you want your bot to come on IRC. Not loading this is equivalent
# to the old NO_IRC define.
loadmodule server
# What is your network?
set net-type 1
#nimi asetuksia
set nick "SlaveBot"
set altnick "SlaveBot-"
set realname "Maaritin testibotti"
# This is a Tcl script to be run immediately after connecting to a server.
bind evnt - init-server evnt:init_server
proc evnt:init_server {type} {
    global botnick
    putquick "MODE $botnick +i-ws"
}
# Set the default port which should be used if none is specified with
# '.jump' or in 'set servers'.
set default-port 6667
# This is the bot's server list.
# The format is:
#  server[:port[:password]]
```

```
set servers {
    irc.elisa.fi
}

# Kohdassa olevat tiedot ovat toimivat perusasetuksilla
#### MODULE ####

# Kohdassa olevat tiedot ovat toimivat perusasetuksilla
loadmodule ctcp
loadmodule irc
loadmodule notes

# Set here the filename where private notes between users are stored.
set notefile "slavebot.notes"

# Kohdassa olevat tiedot ovat toimivat perusasetuksilla
# Kommentoi die komento pois tai asetustiedosto ei toimi
# Comment out this next line. Otherwise, your bot won't start.
#die "You didn't edit your config file completely like you were told, did you?"
#### MODULE ####

# Kohdassa olevat tiedot ovat toimivat perusasetuksilla
loadmodule console
loadmodule uptime
##### SCRIPTS #####

# Kohdassa olevat tiedot ovat toimivat perusasetuksilla
source scripts/alltools.tcl
source scripts/action.fix.tcl
source scripts/dccwhois.tcl
source scripts/userinfo.tcl
loadhelp userinfo.help
# Lopuksi omat skriptit

source scripts/test.tcl
source scripts/lomat.tcl
source scripts/todo.tcl
```

"Mitä tarvitsee tehdä"- skripti

TODO -skripti eggdroppiin. Pohjana käytetty "Dekadent Learn v1.0.1 by CoolMaster for eggdrop" -skriptiä. [15]

Configuration

#Don't forget, all this files need to have +rw permissions you can do that with chmod command.

#The file to store learn data

set learn(file) "learn.dat"

#The temp file when you need to delete/insert something

set learn(tempfile) "learn.tmp"

Triggers

#To delete all tasks --> trigger <nick>

set learn(forget) "!todo_tuhoa"

#To insert a task for user, the task will be added to the end --> trigger <nick>

<definition>

set learn(insert) "!todo"

#To delete one task from user --> trigger <nick> <n>

set learn(del) "!todo_poista"

#This is used to retrieve a list of tasks for user --> trigger <nick>

set learn(view) "!todo_lista"

#The list of tasks in learn file

set learn(wordlist) "!todo_listat"

#The help with all triggers and other misc stuff.

set learn(help) "!todo_help"

Misc

#This is the method PRIVMSG/NOTICE for user when something changed

set learn(method) "NOTICE"

#This is the method PRIVMSG/NOTICE when you try to retrieve a definition
set learn(method-def) "PRIVMSG"

Don't change nothing below if you don't know what you are
doing #####

BINDS

bind pub - \$learn(forget) learn_forget

bind pub - \$learn(insert) learn_insert

bind pub - \$learn(del) learn_del

bind pub - \$learn(view) learn_view

bind pub - \$learn(wordlist) learn_list

bind pub - \$learn(help) learn_help

Misc

putlog "Loaded TODO"

if {[string compare \$learn(file) \$learn(tempfile)] == 0} {die "Error! You have the same
name to other types of file! See config."}

if {[file exists \$learn(file)]} {set fxtmp [open \$learn(file) w] ; close \$fxtmp}

Procs

Verify if task already exists, if yes will return 1 else return 2

proc check_word {word} {

global learn

set fx [open \$learn(file) r]

while {[!eof \$fx]} {

set word_check [lindex [gets \$fx] 0]

if {[string compare \$word \$word_check] == 0} {close \$fx ; return 1}

}

close \$fx ; return 2

}

```
proc learn_add {nick host hand chan text} {  
    global learn  
    set word [string tolower [lindex $text 0]]  
    set definition [lrange $text 1 end]  
    if {$definition == ""} {  
        putserv "$learn(method) $nick :Syntaksivirhe, käytä: $learn(add) <word>  
<definition>"  
        return 0  
    }  
    if {[check_word $word] == 1} {  
        putserv "$learn(method) $nick :Tehtävä on jo olemassa"  
        return 0  
    }  
    set fx [open $learn(file) a]  
    puts $fx "$word $definition |"  
    close $fx  
    putserv "$learn(method) $nick :Lisätty \"$word\" tehtäviin"  
}
```

```
proc learn_forget {nick host hand chan text} {  
    global learn  
    set word [string tolower [lindex $text 0]]  
    if {$word == ""} {  
        putserv "$learn(method) $nick :Syntaksivirhe, käytä: $learn(forget) <word>"  
        return 0  
    }  
    if {[check_word $word] == 2} {  
        putserv "$learn(method) $nick :Ei pystytty poistamaan: tehtävää ei ole"  
        return 0  
    }  
    set fx1 [open $learn(file) r]  
    set fx2 [open $learn(tempfile) w]
```

```
while {[eof $fx1]} {  
    set tmp [gets $fx1]  
    set word_check [lindex $tmp 0]  
    if {$tmp != ""} {if {[string compare $word $word_check] == 0} {continue} else  
{puts $fx2 $tmp}}  
}  
close $fx1 ; close $fx2  
exec rm -f $learn(file) ; exec mv $learn(tempfile) $learn(file)  
putserv "$learn(method) $nick :\"$word\" poistettu tehtävistä"  
}
```

```
proc learn_list {nick host hand chan text} {  
    global learn  
    set words "" ; set check 0  
    set fx [open $learn(file) r]  
    while {[eof $fx]} {  
        set tmp [lindex [gets $fx] 0]  
        set words "$words $tmp"  
        if {$tmp != ""} {set check 1}  
    }  
    close $fx  
    if {$check == 0} {putserv "$learn(method) $nick :Ei tehtäviä"} else {putserv  
"$learn(method) $nick :Tehtävät: $words"}  
}
```

```
proc learn_view {nick host hand chan text} {  
    global learn  
    set word [string tolower [lindex $text 0]]  
    if {$word == ""} {  
        putserv "$learn(method) $nick :Syntaksivirhe, käytä: $learn(view) <word>"  
        return 0  
    }  
    if {[check_word $word] == 2} {
```



```

    putserv "$learn(method) $nick :Tehtäviä ehdolla \"$word\" ei ole tiedossa"
    return 0
}
set fx [open $learn(file) r]
while {[eof $fx]} {
    set tmp [gets $fx]
    if {[string compare $word [lindex $tmp 0]] != 0} {continue}
    set i 1
    set defs [split [lrange $tmp 1 end] "|"]
    set defi ""
    while {1} {
        set d [lindex $defs [expr $i-1]]
        if {$d == ""} {break}
        putserv "$learn(method-def) $chan :$word\002\[002$i\002\]\002\037:\037 $d"
        incr i 1
    }
}
close $fx
}

proc learn_del {nick host hand chan text} {
    global learn
    set word [string tolower [lindex $text 0]]
    set n [lindex $text 1]
    if {$n == ""} {
        putserv "$learn(method) $nick :Syntaksivirhe, käytä: $learn(del) <word> <N>"
        return 0
    }
    if {[check_word $word] == 2} {
        putserv "$learn(method) $nick :Sanaa ei löydy"
        return 0
    }
    set fx1 [open $learn(file) r]

```

```

set fx2 [open $learn(tempfile) w]
while {[eof $fx1]} {
    set tmp [gets $fx1]
    set word_check [lindex $tmp 0]
    set i 0
    set defs [split [lrange $tmp 1 end] "|"]
    set defi ""
    while {1} {
        if {[expr $i+1] == $n} {incr i 1 ; continue}
        set d [lindex $defs $i]
        if {$d == ""} {break}
        if {$defi == ""} {set defi $d;} else {set defi $defi$d;}
        incr i 1
    }
    if {$tmp != ""} {
        if {[string compare $word $word_check] == 0} {puts $fx2 "$word_check $defi"}
    } else {puts $fx2 $tmp}
    }
}
close $fx1 ; close $fx2
exec rm -f $learn(file) ; exec mv $learn(tempfile) $learn(file)
putserv "$learn(method) $nick :Tehtävä \"$n\" poistettu"
}

```

```

proc learn_insert {nick host hand chan text} {
    global learn
    set word [string tolower [lindex $text 0]]
    set definition [lrange $text 1 end]
    if {$definition == ""} {
        putserv "$learn(method) $nick :Syntaksivirhe, käytä: $learn(insert) <word>
<definition>"
        return 0
    }
}

```

```
if {[check_word $word] == 2} {  
    learn_add $nick $host $hand $chan $text  
    return 0  
}  
set fx1 [open $learn(file) r]  
set fx2 [open $learn(tempfile) w]  
while {![eof $fx1]} {  
    set tmp [gets $fx1]  
    set word_check [lindex $tmp 0]  
    if {$tmp != ""} {  
        if {[string compare $word $word_check] == 0} {puts $fx2 "$tmp$definition |"}  
    }  
    else {puts $fx2 $tmp}  
}  
close $fx1 ; close $fx2  
exec rm -f $learn(file) ; exec mv $learn(tempfile) $learn(file)  
putserv "$learn(method) $nick :Tehtävä lisätty"  
}  
  
proc learn_help {nick host hand chan text} {  
    global learn  
    putserv "$learn(method) $nick :\002*** Yritysten oma Botti ***\002"  
    putserv "$learn(method) $nick :\"!todo_lista nick\" näyttää listan nickin todo-listasta"  
    putserv "$learn(method) $nick :\"!todo nick asia\" lisää nickille "  
    putserv "$learn(method) $nick :\"!todo_poista nick numero\" poistaa nickiltä listalta  
    sen numeron mikä numero kohtaan on annettu"  
    putserv "$learn(method) $nick :\"!todo_tuhoa nick\" tuhoaa koko nickin listan"  
    putserv "$learn(method) $nick :\"!todo_listat\" näyttää botilla olevat nickit"  
}
```